

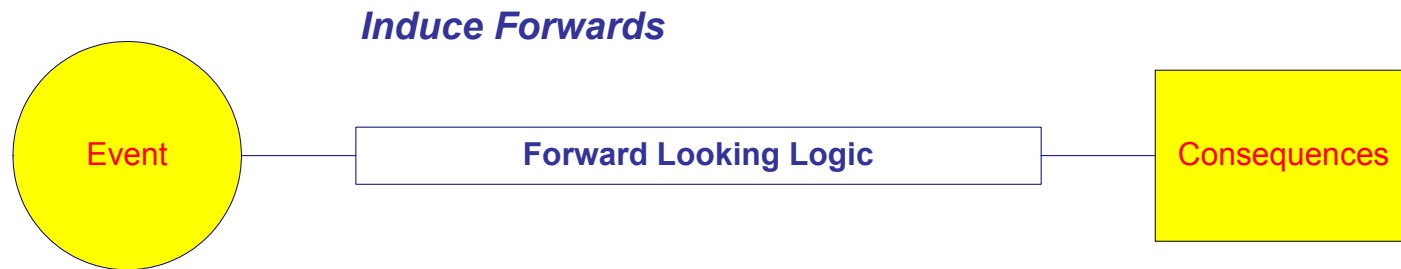
Fault Tree Analysis (FTA): Concepts and Applications

Bill Vesely
NASA HQ

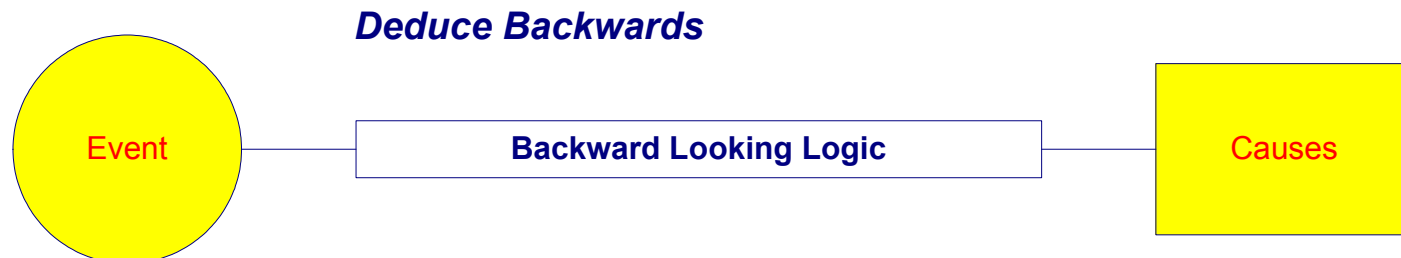


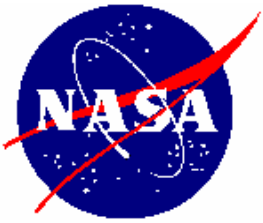
Inductive and Deductive Modeling are the Two Basic Types of Modeling

- **Inductive** models forwardly *induce* the consequences of an event.



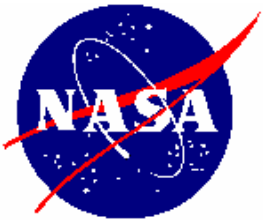
- **Deductive** models backwardly *deduce* the causes of an event.





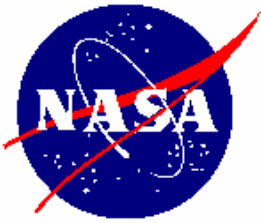
An Inductive Model Defines Scenarios for an Initiating Event

- **An initiating event is first defined which can have undesired consequences.**
- **Subsequent events are identified which define possible progressions of the initiating event.**
- **Possible realizations of the subsequent events are defined and linked to model scenarios.**
- **The consequence of each scenario is described.**



A Deductive Model Resolves the Causes for an Event

- **An event is first defined for which causes are to be resolved.**
- **The event is resolved into its immediate and necessary sufficient causal events.**
- **The event is related to the causal events using appropriate logic.**
- **This stepwise resolution of events into immediate causal events proceeds until basic causes (primary causes) are identified.**



Fault Tree Analysis: a Systematic and Stylized Deductive Process

- An *undesired event* is defined
- The event is resolved into its *immediate causes*
- This resolution of events continues until *basic causes* are identified
- A logical diagram called a *fault tree* is constructed showing the logical event relationships



Benefits of Constructing a Fault Tree

- The fault tree explicitly shows all the different relationships that are necessary to result in the top event
- In constructing the fault tree, a thorough understanding is obtained of the logic and basic causes leading to the top event
- The fault tree is a tangible record of the systematic analysis of the logic and basic causes leading to the top event
- The fault tree provides a framework for thorough qualitative and quantitative evaluation of the top event



Elements of Fault Tree Analysis (FTA)

- FTA is a deductive analysis approach for resolving an undesired event into its causes
- FTA is a backward looking analysis, looking backward at the causes of a given event
- Specific stepwise logic is used in the process
- Specific logic symbols are used to to illustrate the event relationships
- A logic diagram is constructed showing the event relationships.



Why FTA is carried out

- To exhaustively identify the causes of a failure
- To identify weaknesses in a system
- To assess a proposed design for its reliability or safety
- To identify effects of human errors
- To prioritize contributors to failure
- To identify effective upgrades to a system
- To quantify the failure probability and contributors
- To optimize tests and maintenances



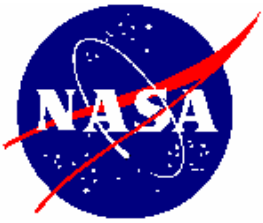
Role of FTA in System Safety Analysis

- **FTA is used to resolve the causes of system failure**
- **FTA is used to quantify system failure probability**
- **FTA is used to evaluate potential upgrades to a system**
- **FTA is used to optimize resources in assuring system safety**
- **FTA is used to resolve causes of an incident**
- **FTA is used to model system failures in risk assessments**



Role of FTA in PRA

- **A Probabilistic Risk Assessment (PRA) models event scenarios**
- **An event scenario consists of an initiating event and subsequent system failures**
- **FTA is carried out to model the causes of the system failures**
- **Using data on the probability of the causes, the probability of system failure is determined**
- **The probability of the accident scenario is thereby determined**



The Thought Process in FTA

- FTA is backward looking
- The end result is the analysis starting point
- The end result is then traced back one step at a time to its immediate causes
- The relationships of the causes, or events, are shown with logic symbols
- This backward tracing process continues until the basic causes are identified
- FTA systematizes and codifies the process



Comparison of FTA with Other Approaches

- FTA is not a Fishbone analysis which is a more informal depiction of event causes (**informal deductive**)
- FTA is not an FMEA which assesses different effects of single basic causes (**inductive**)
- FTA is not Event Tree Analysis which assesses the consequences of given initiating events (**inductive**)
- FTA is a formal approach for resolving the basic causes of a given undesired event (**formal deductive**)

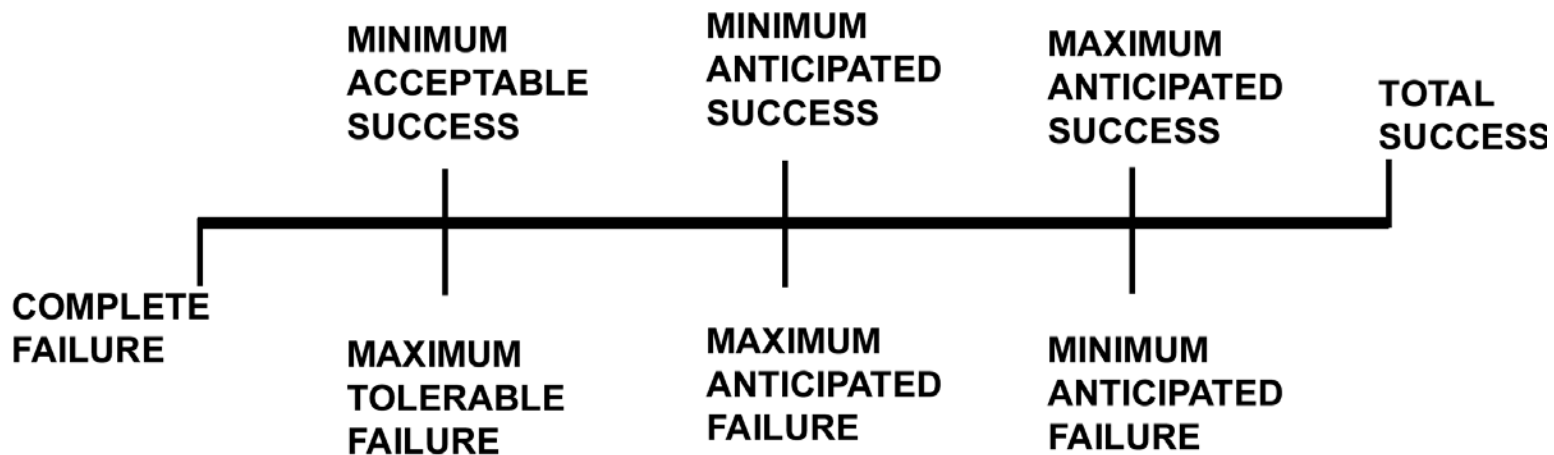


FTA Operates in Failure Space

- **Designers design for success**
- **Safety analysts analyze for failure**
- **There can be various degrees of success**
- **Thresholds for failure are identifiable**
- **Failure events can be more readily discretized**
- **Failure quantifications are simpler**
- **The “failure mindset” probes for weaknesses and gaps**

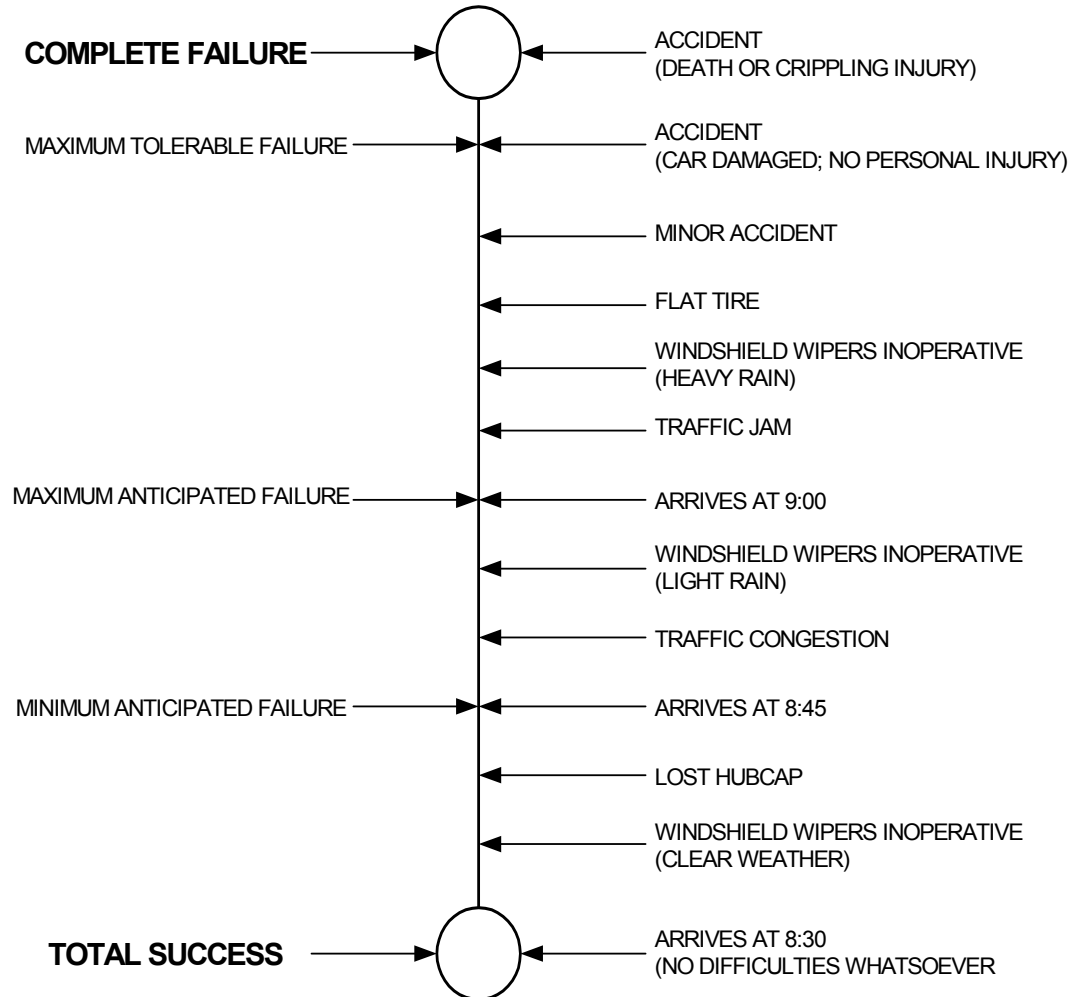


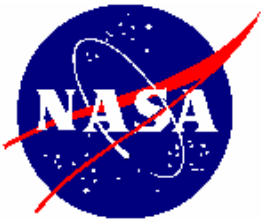
Success Space Versus Failure Space





Different Failure and Success States for a Trip





A Fault Tree Models Failure Modes

- A failure mode is the failure state of the system or component
- Examples of failure modes are fail to start, fail to open, fail to shutdown
- In contrast, failure mechanisms are the processes by which failures occur
- Examples of failure mechanisms are corrosion, overpressure, and fatigue
- A failure mechanism is only included in the failure mode definition when detailed mechanisms are modeled

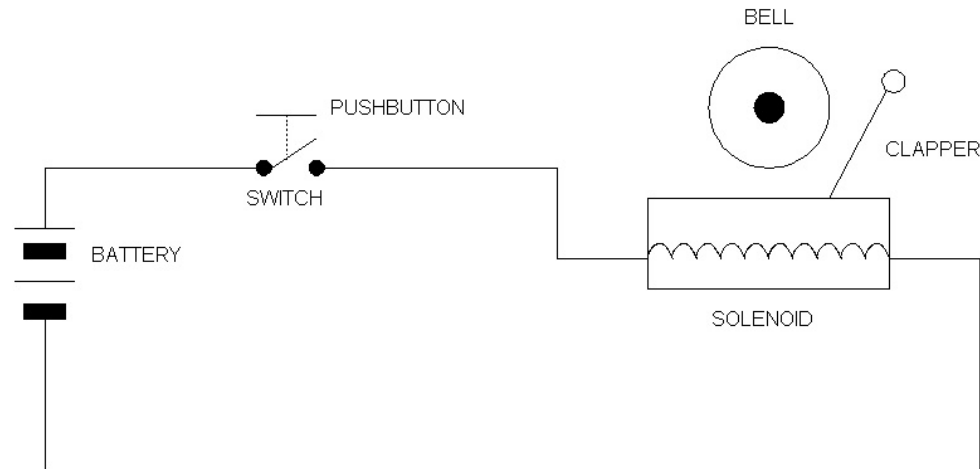


Illustration of Failure Mode Versus Failure Mechanism

Description of Event	System	Subsystem	Valve	Actuator
No flow from subsystem when required	Mechanism	Mode	Effect	
Valve unable to open		Mechanism	Mode	Effect
Binding of actuator stem			Mechanism	Mode
Corrosion of actuator stem				Mechanism



Door Bell Example Differentiating Failure Modes and Failure Mechanisms





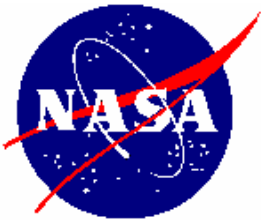
Failure Modes and Mechanisms of the Door Bell System

Failure Effect	Failure Mode	Mechanism
Switch fails to make contact	<ul style="list-style-type: none">▪ Contacts broken▪ High contact resistance	<ul style="list-style-type: none">▪ Mechanical shock▪ Corrosion
Bell-solenoid unit fails to ring	<ul style="list-style-type: none">▪ Clapper broken or not attached▪ Clapper stuck▪ Solenoid link broken or stuck▪ Insufficient magneto-motive force	<ul style="list-style-type: none">▪ Shock▪ Corrosion▪ Open circuit in solenoid▪ Short circuit in solenoid
Low voltage from battery	<ul style="list-style-type: none">▪ No electrolyte▪ Positive pole broken	<ul style="list-style-type: none">▪ Leak in casing▪ Shock



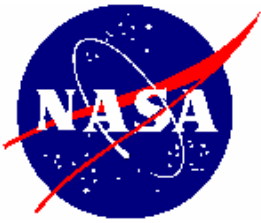
Failure Mechanisms and Failure Causes

- In some areas, failure mechanism and failure cause are differentiated
- A failure cause is defined as the initiator of a failure (example: valve fails to open because of stuck operator)
- A failure mechanism is defined as the process by which the failure occurs (e.g. a valve fails to open because of a stuck operator due to corrosion buildup)
- In FTA, what is important is that the failure mode be precisely define which is What and When describing the fault or failure



Review Questions

1. When should inductive modeling be considered?
2. When should deductive modeling be considered?
3. What are the advantages of working in failure space? Could we develop success-based models?
4. What characterizes FTA as a distinct, deductive modeling approach?
5. Can failure modes, failure mechanisms, and failure causes be defined at different levels?
6. Consider the Main Engine of the Space Shuttle. What are possible failure modes, failure causes and failure mechanisms?

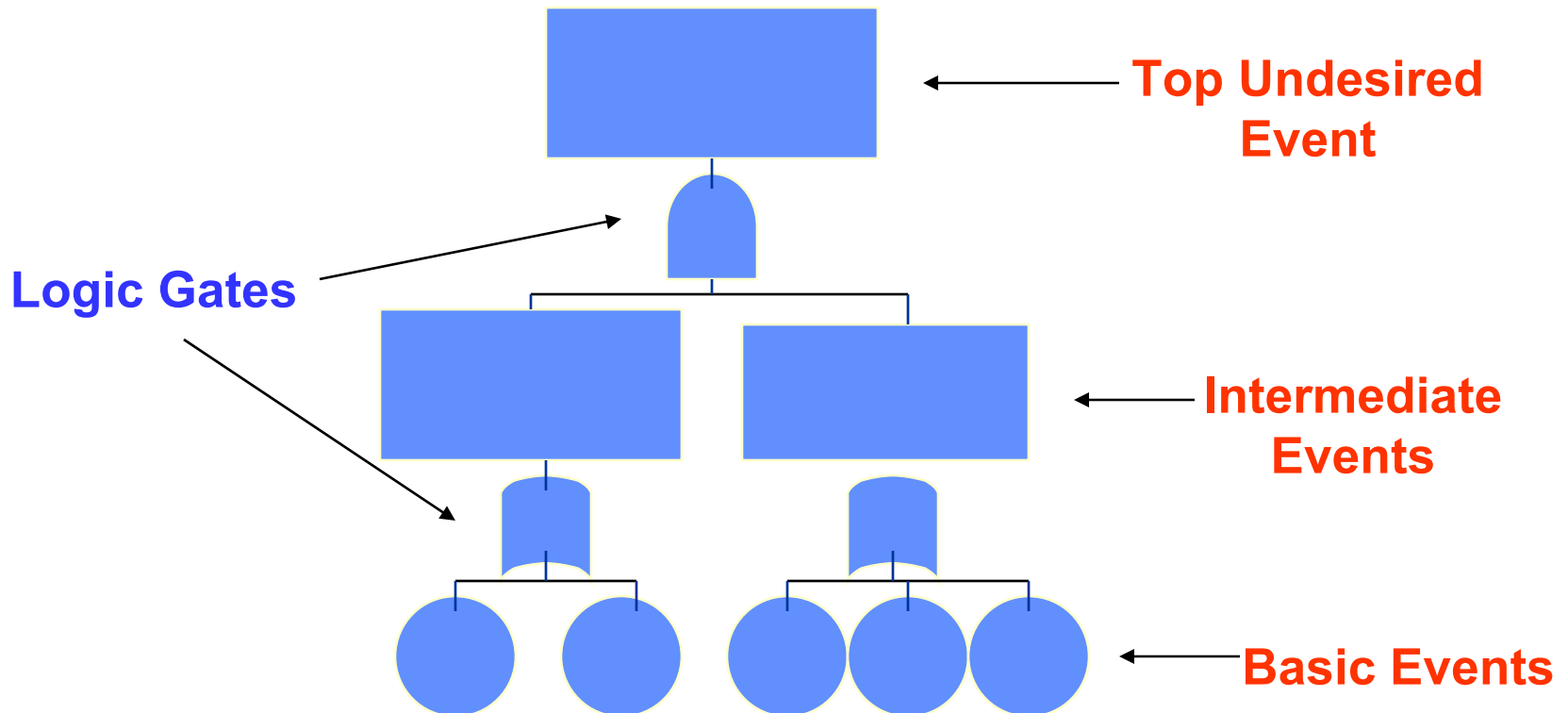


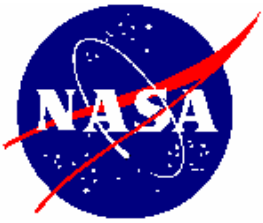
The Fault Tree

- FTA produces a *Fault Tree*
- The fault tree is the *logical model* of the relationship of the undesired event to more basic events.
- The *top event* of the fault tree is the undesired event.
- The *middle events* are intermediate events.
- The bottom of the fault tree is the causal *basic events* or *primary events*.
- The logical relationships of the events are shown by logical symbols or *gates*.



Basic Fault Tree Structure





The Four Necessary Steps to Begin a Fault Tree

1. Define the undesired event to be analyzed (the **focus** of the FTA)
2. Define the boundary of the system (the **scope** of the FTA)
3. Define the basic causal events to be considered (the **resolution** of the FTA)
4. Define the **initial state** of the system

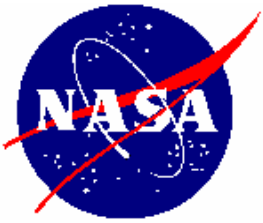
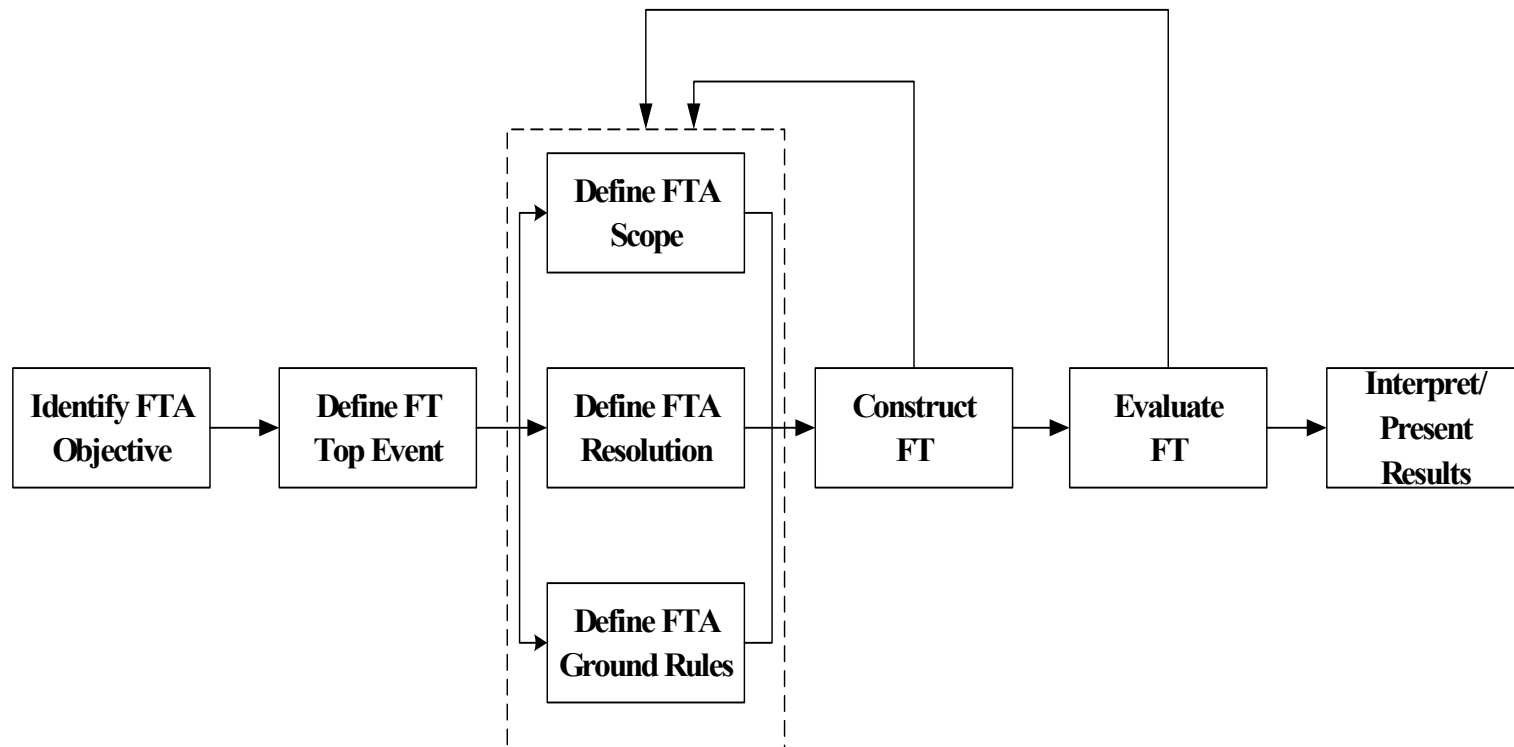


Illustration of the Steps of a FTA

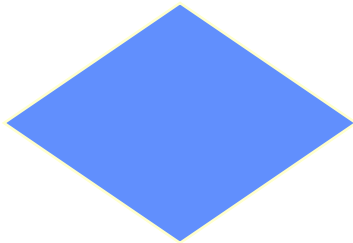




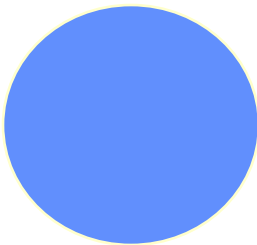
Basic Events of a Fault Tree



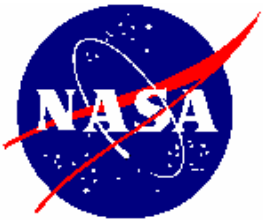
Top Event or Intermediate Event



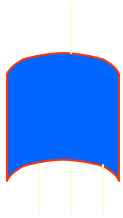
Undeveloped Event



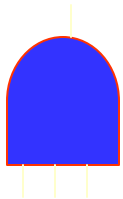
Basic Event



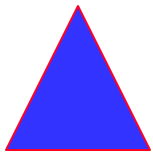
Basic Gates of a Fault Tree



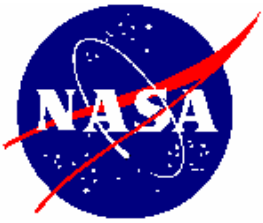
OR gate - the above output event occurs if **either** of the input lower level events occur



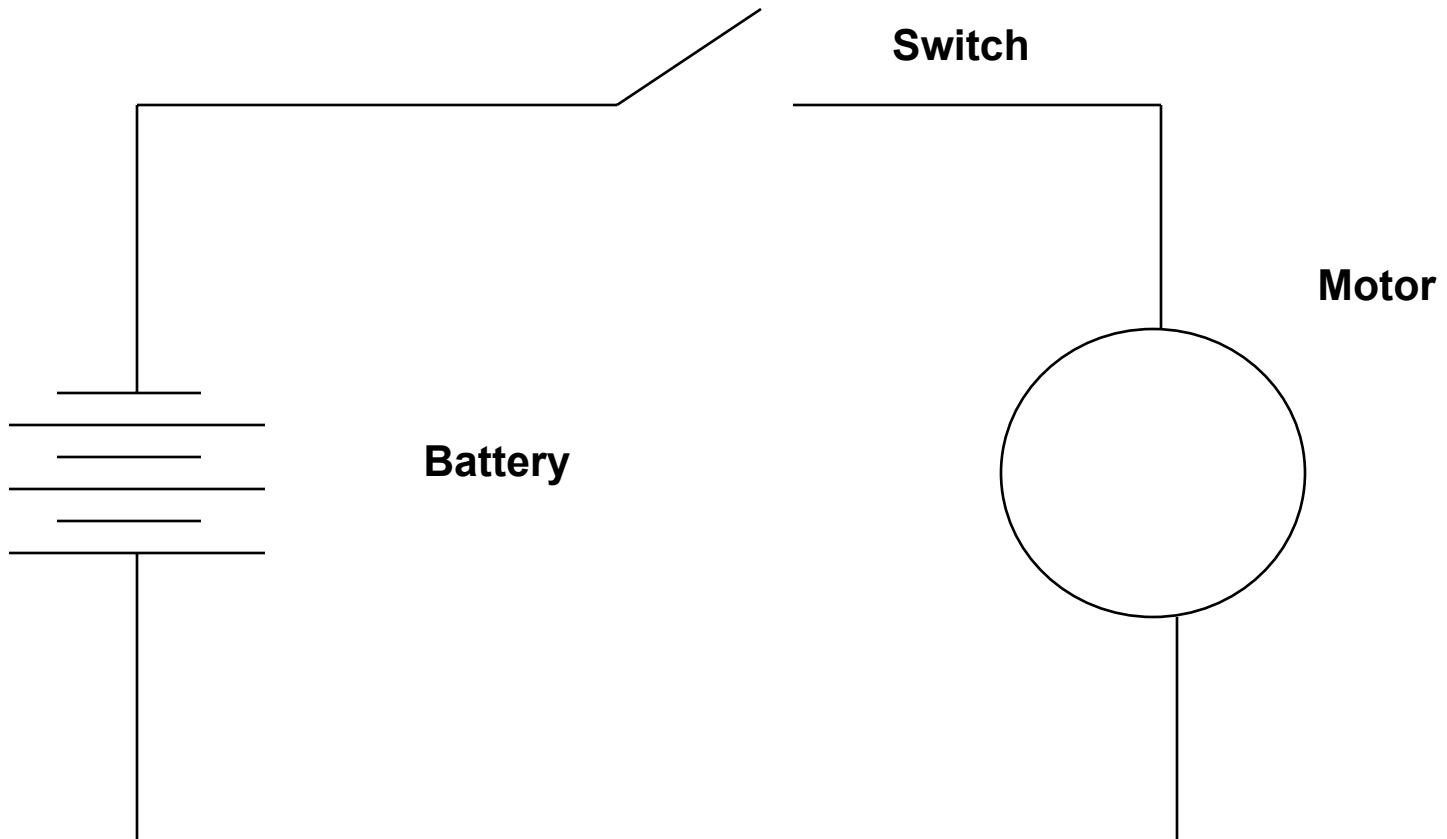
AND gate - the above output event occurs if **all** of the input lower level events occur

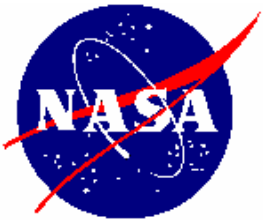


TRANSFER gate transfer to/from another part of the fault tree



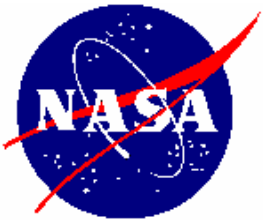
Simple Battery Powered Circuit (BPC)



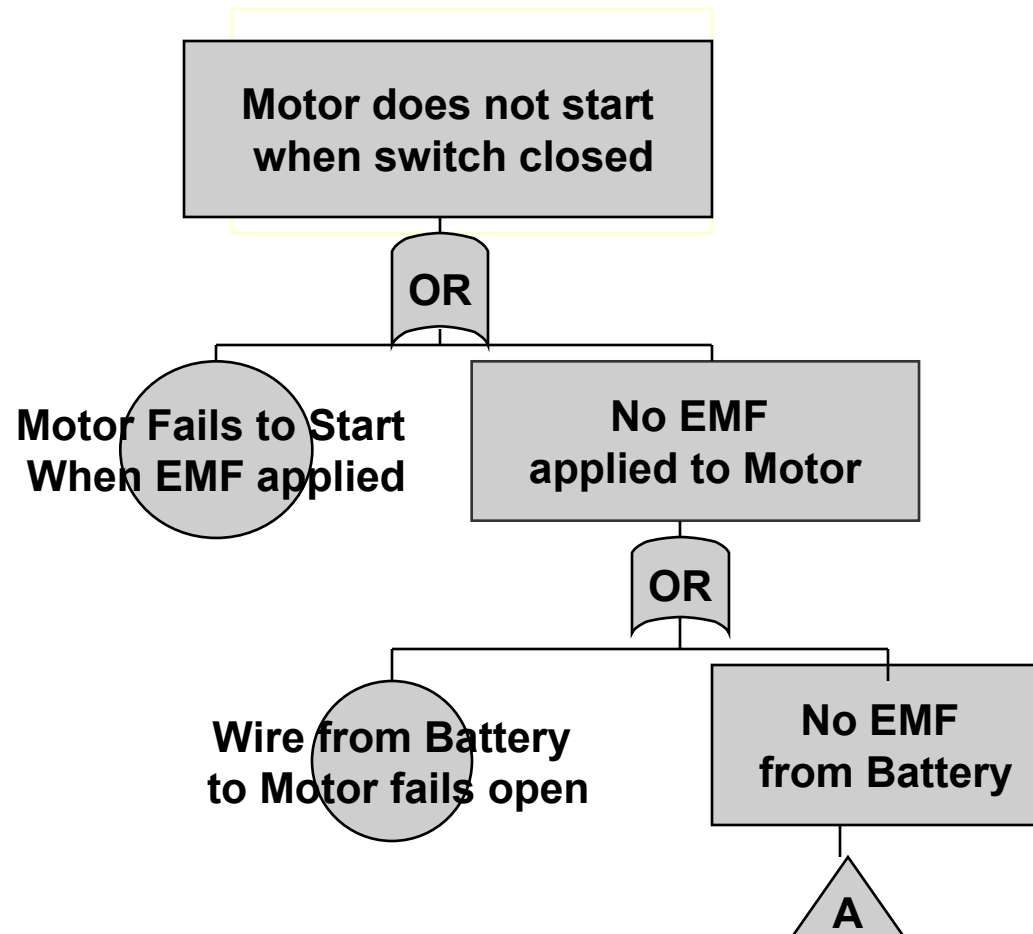


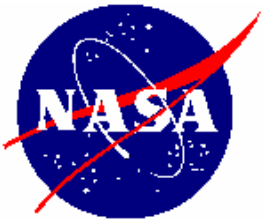
Specifications for the BPC FT

- ***Undesired top event:*** Motor does not start when switch is closed
- ***Boundary of the FT:*** The circuit containing the motor, battery, and switch
- ***Resolution of the FT:*** The basic components in the circuit excluding the wiring
- ***Initial State of System:*** Switch open, normal operating conditions

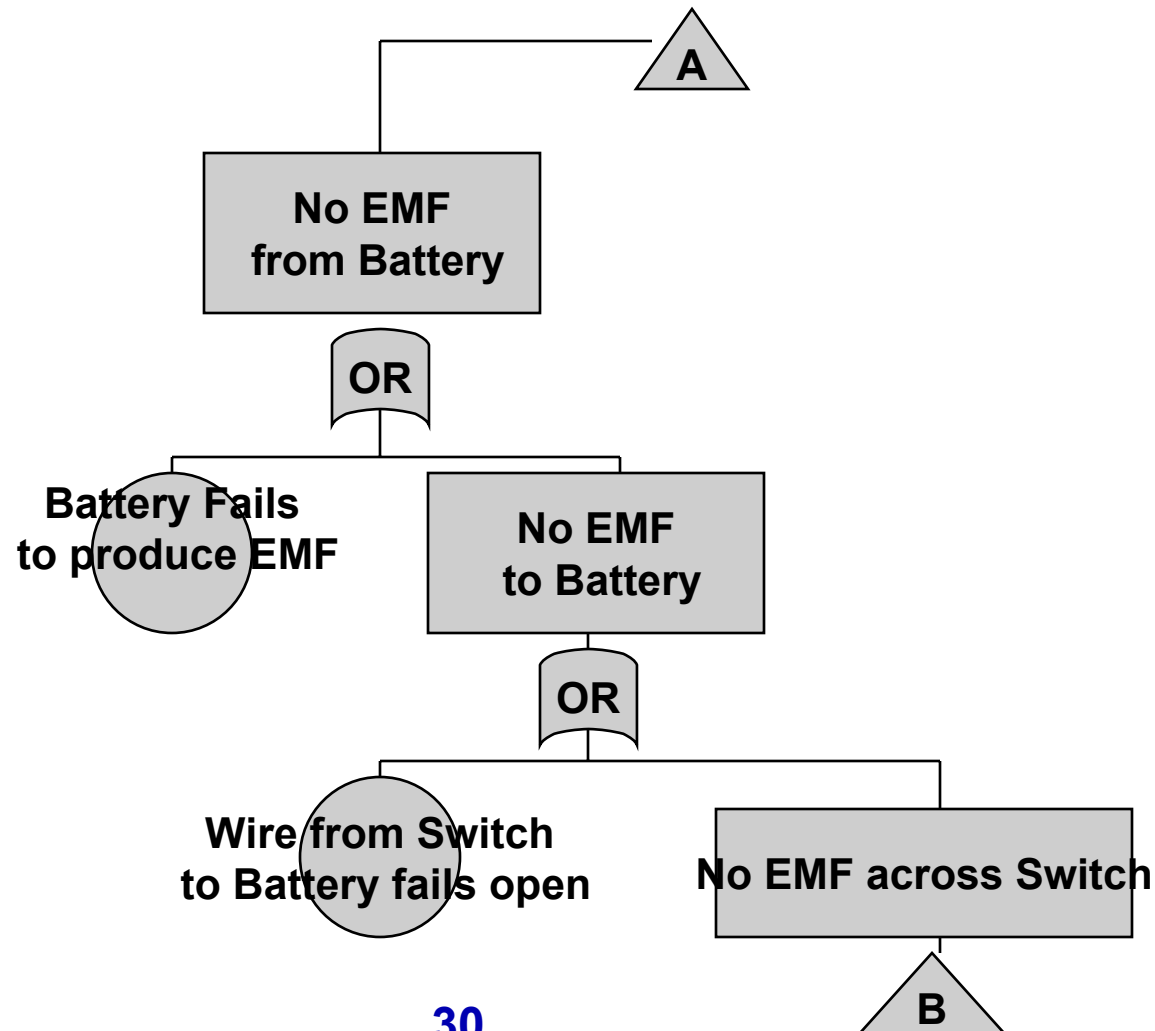


Start of BPC FT (1)



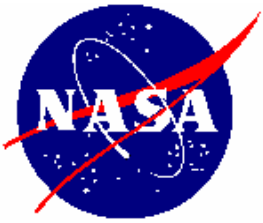


Continuation of the BPC FT (2)



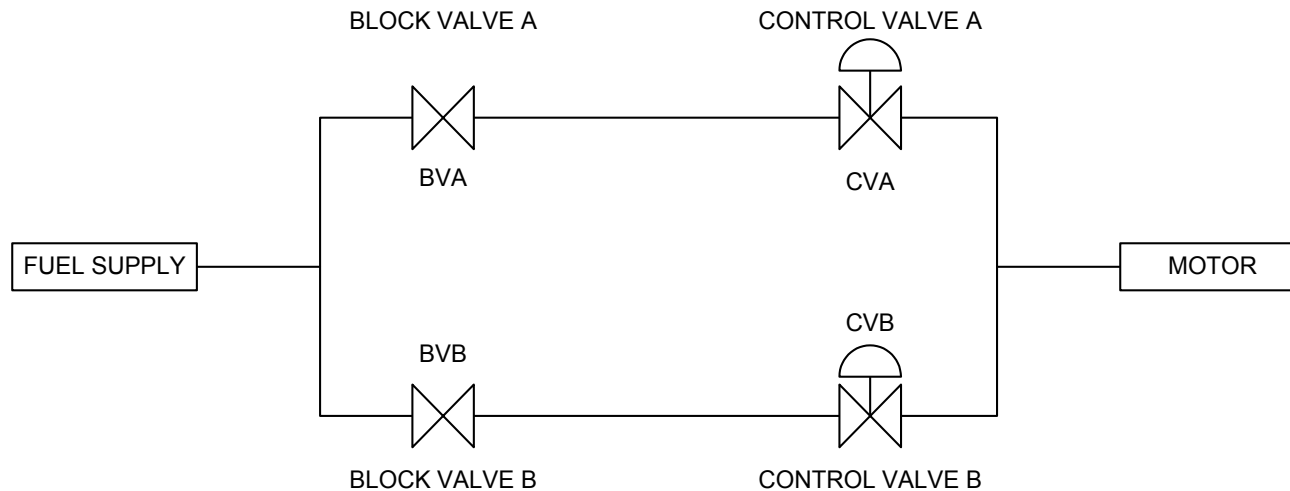


The diagram is a fault tree for the event "No EMF across Switch". The top event is represented by a rectangle labeled "No EMF across Switch". This event is connected to a central "OR" gate, represented by a rounded rectangle labeled "OR". The "OR" gate has two inputs, each represented by a circle. The left input circle is labeled "Switch fails to contact". The right input circle is labeled "Wire from Switch to Motor fails open". A line from the top of the "No EMF across Switch" rectangle extends upwards and then to the right, ending at a triangle labeled "B", which represents the basic event "B".



Fault Tree Exercise: Fuel Supply System

Top Event: No Fuel to Motor When Requested



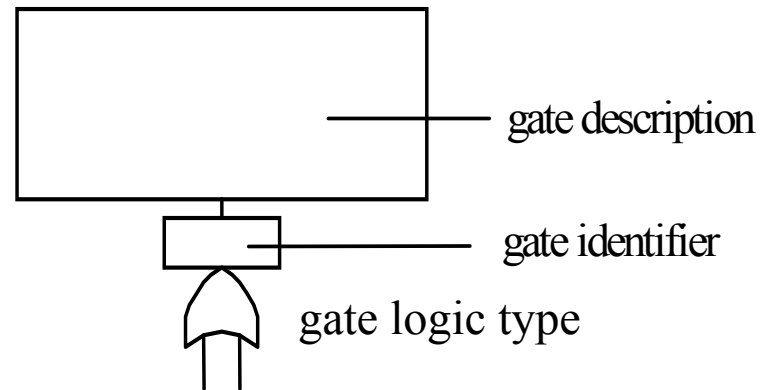
Control valves: initially closed, opened manually

Block valves normally open

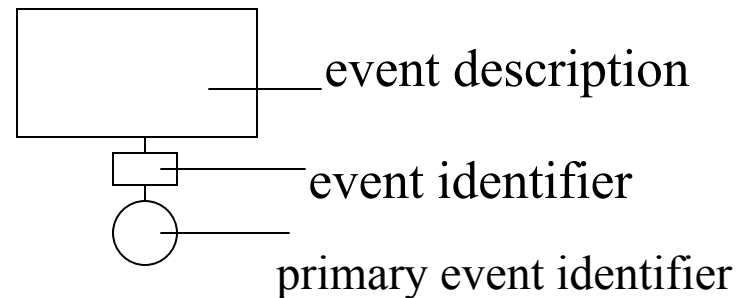


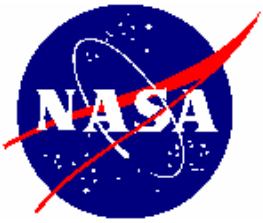
Symbols Used in FTA Software Programs

Intermediate Event (Gate)

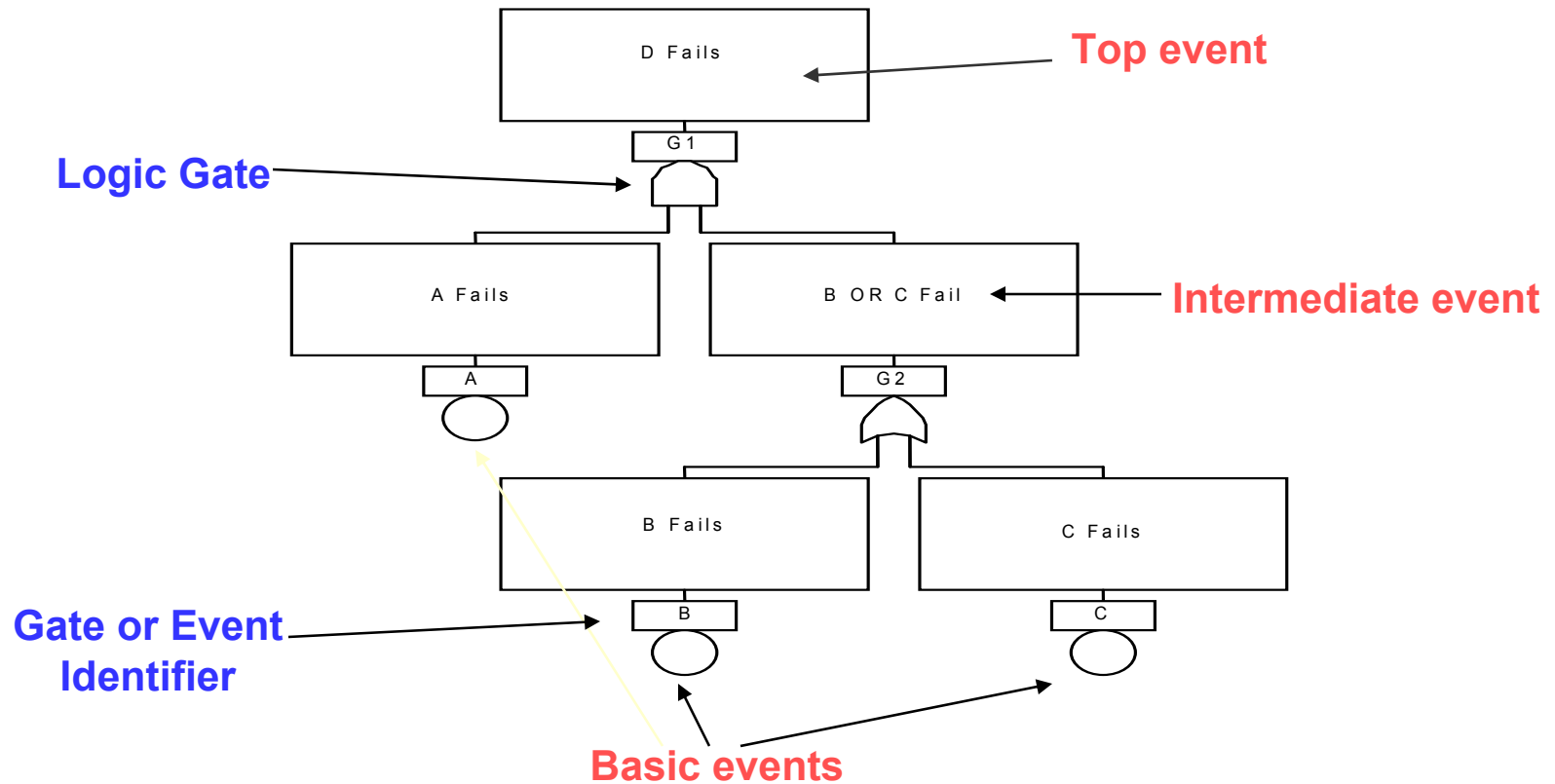


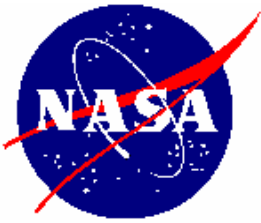
Primary Event (Basic Cause)



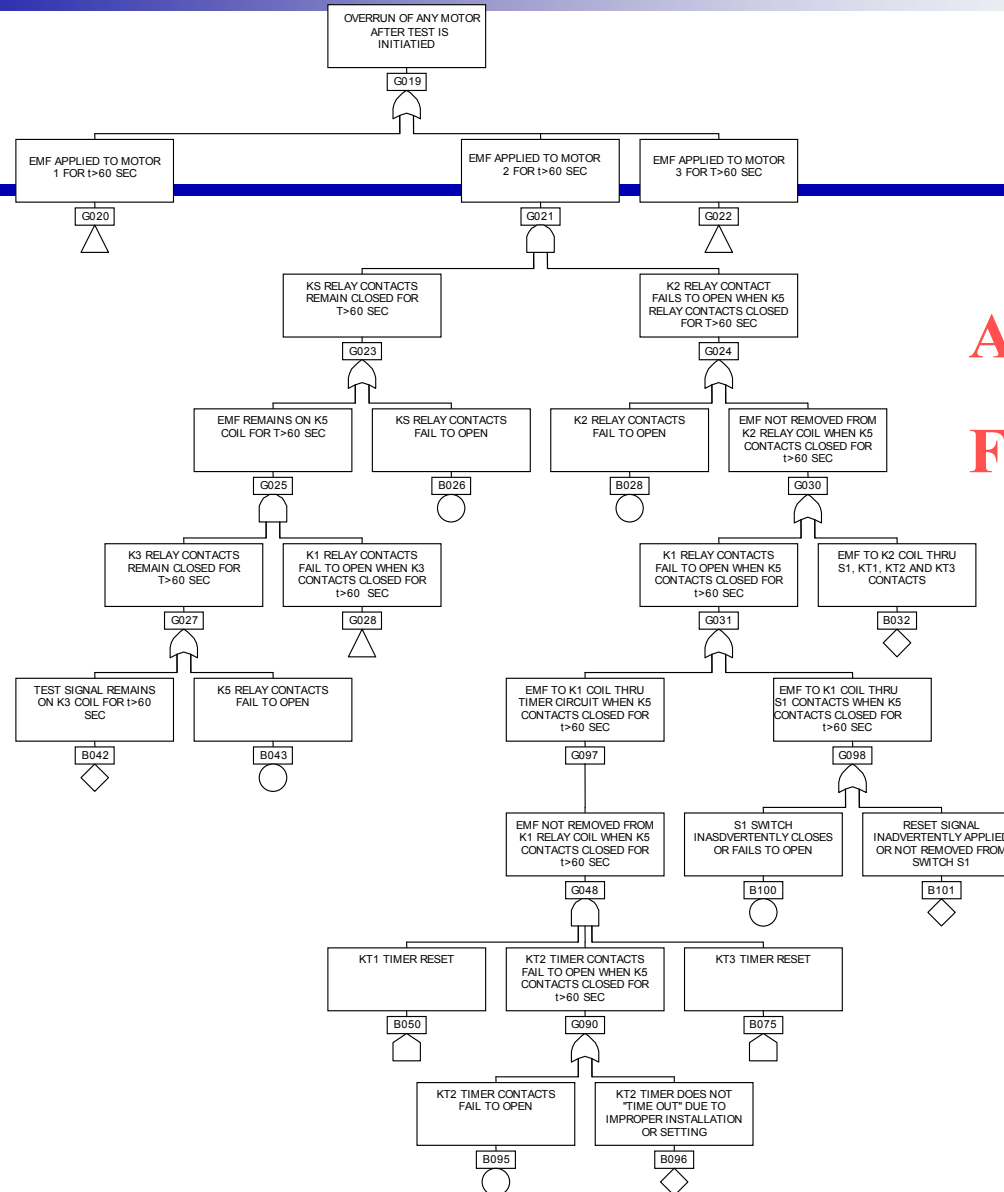


Fault Tree Software Representations





Mission Success Starts With Safety



A Typical
Fault Tree



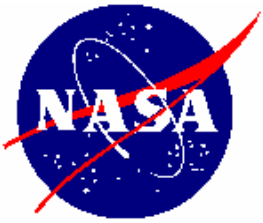
The Top Event of the Fault Tree

- The top event should describe **WHAT** the event is and **WHEN** it happens
- The top event is often a system failure but can be any other event
- The top event is the specific event to be resolved into its basic causes
- Defining the wrong top event will result in wrong assessments and conclusions



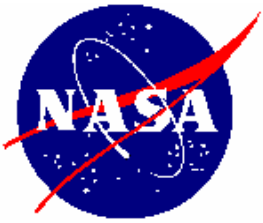
Examples of Top Event Definitions

- Fire Suppression System **Fails to Operate** when actuated
- Fire Suppression System **Inadvertently Activates** during normal conditions
- Auxiliary Power System **Fails to Continually Operate** for the required time period
- Fuel Supply System **Fails to Shutoff** after the fueling phase
- Launch Vehicle **Fails to Ignite** at Launch
- Launch Vehicle **Suffers a Catastrophic Failure** at Launch

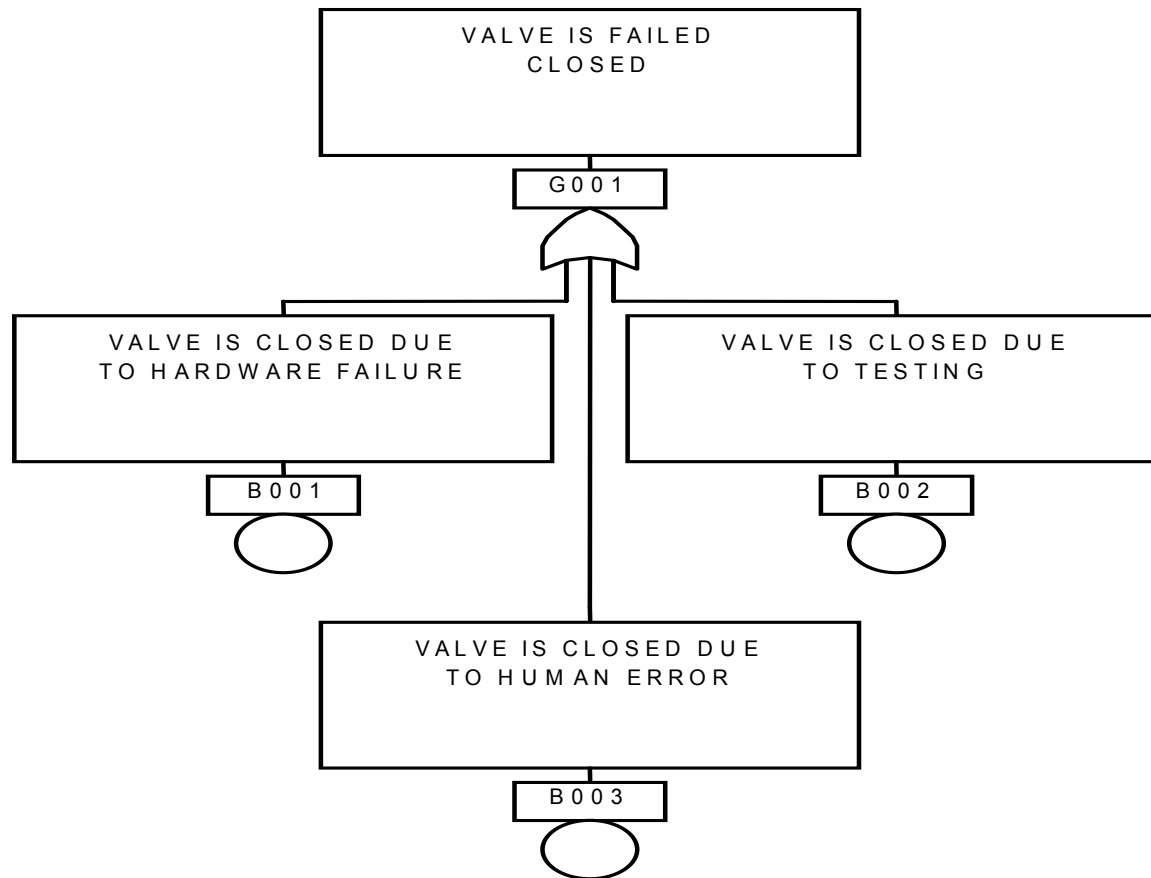


The OR Gate

- The OR Gate represents the **logical union** of the inputs: the output occurs if **any** of the inputs occur
- The OR gate is used when an event is resolved into **more specific** causes or scenarios
- The OR gate is used when a component failure is resolved into an *inherent failure* **or** a *command failure*
- The OR gate is used when an event is described in terms of equivalent, more specific events



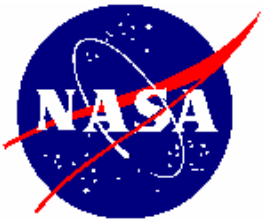
An OR Gate Resolving A Component Failure into Specific Failures



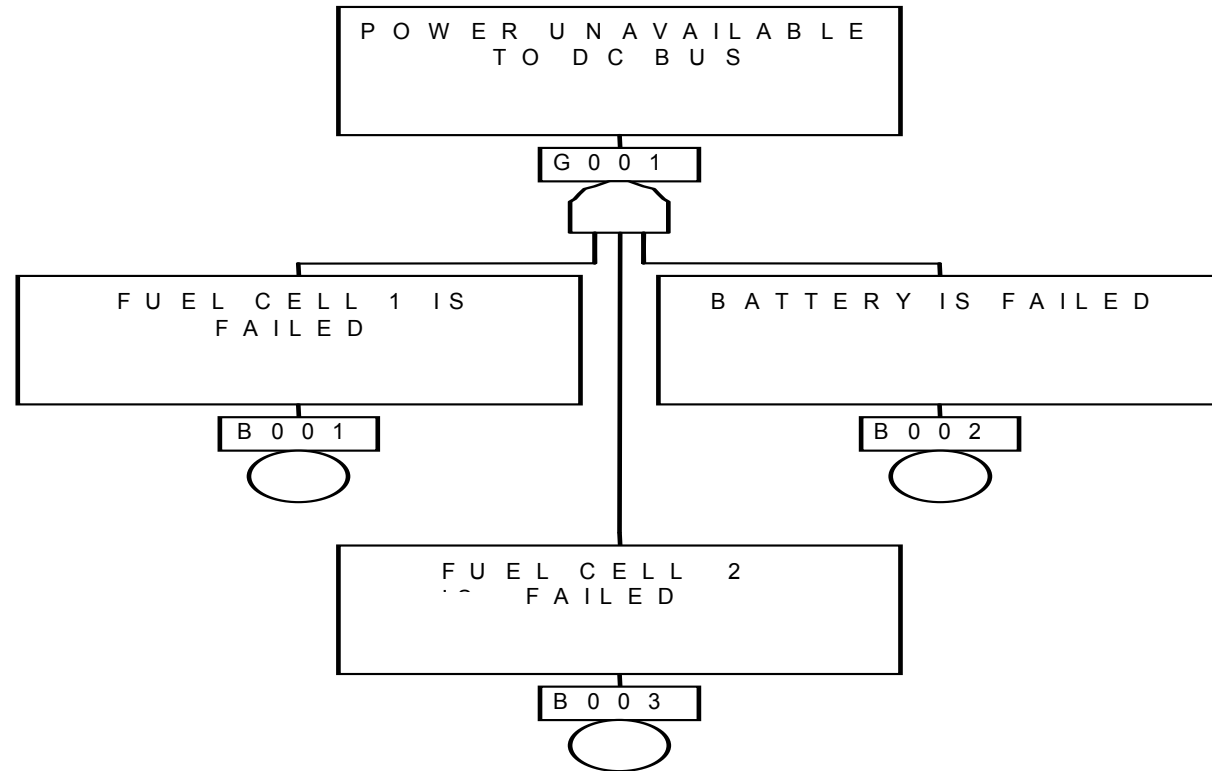


The AND Gate

- The AND Gate represents the **logical intersection** of the inputs: the output occurs if **all** of the inputs occur
- The OR gate is used when an event is resolved into **combinations** of events that need to occur
- The **AND** gate is used when a redundant system is resolved into **multiple** subsystems that need to fail
- The **AND** gate is used when a system failure is resolved into conditions **and** events needed to occur



AND Gate for a Redundant Power Supply

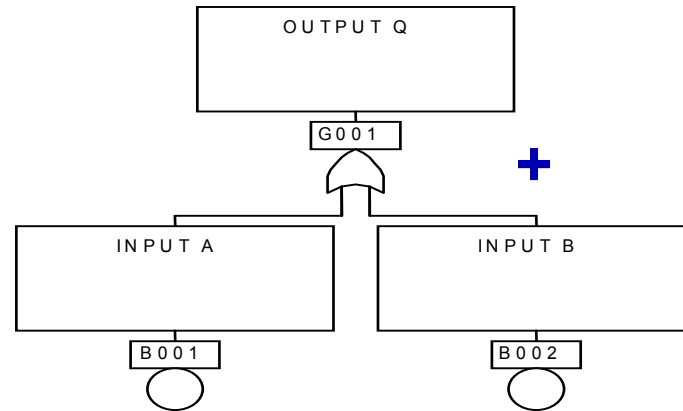




Summary of OR and AND Gates

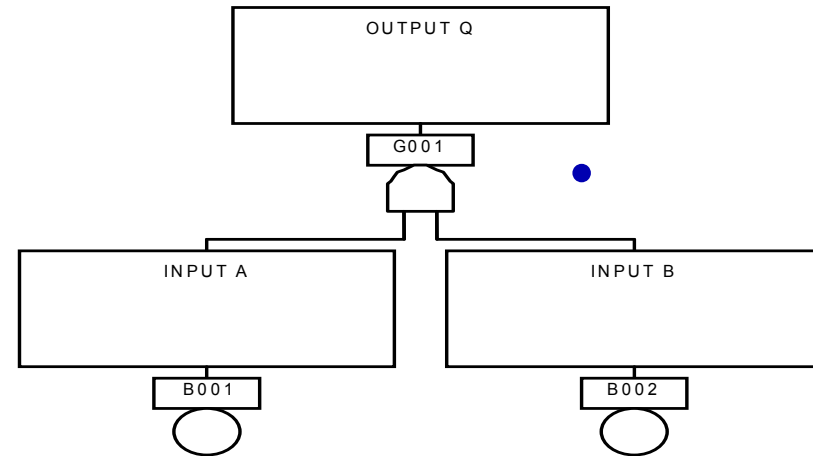
OR Gate

(Logical Plus Gate)



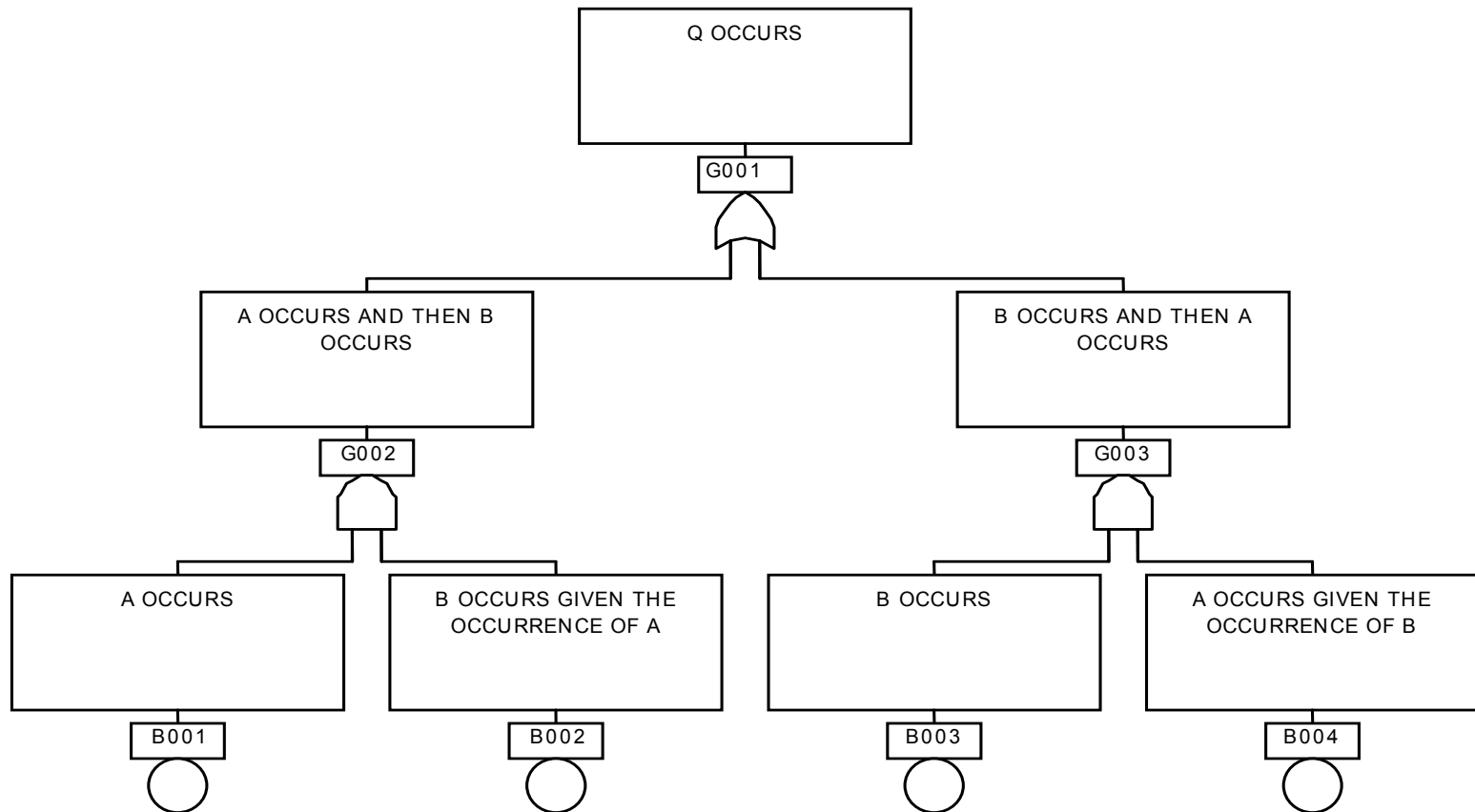
AND Gate

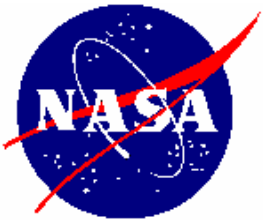
(Multiplication Gate)





Linking OR and AND Gates



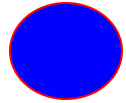


Terminating Events in a Fault Tree

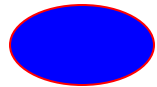
- **The terminating events of a fault tree identify where the FTA stops**
- **Two fundamental terminating events are the basic event and the undeveloped event**
- **The basic event represents the lowest level event (cause) resolved in the fault tree**
- **The undeveloped event represents an event which is not further developed for causes**



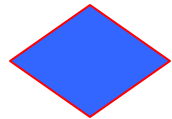
Expanded Types of Terminating Events



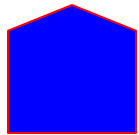
Basic Causal Event- treated as a primary cause with no further resolution



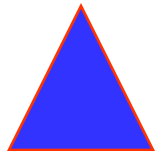
Condition Event- defines a condition which needs to exist



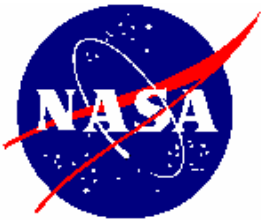
Undeveloped Event- not further developed



House Event- an event expected to occur. Sometimes used as a switch of True or False

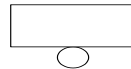


Transfer Symbol- transfer out of a gate or into a gate

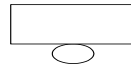


Extended Gate Symbols

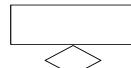
PRIMARY EVENT SYMBOLS



BASIC EVENT - A basic initiating fault requiring no further development



CONDITIONING EVENT - Specific conditions or restrictions that apply to any logic gate (used primarily with PRIORITY AND and INHIBIT gates)



UNDEVELOPED EVENT - An event which is not further developed either because it is of insufficient consequence or because information is unavailable



HOUSE EVENT - An event which is normally expected to occur

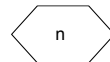
GATE SYMBOLS



AND - Output fault occurs if all of the input faults occur



OR - Output fault occurs if a least one of the input faults occurs



COMBINATION - Output fault occurs if n of the input faults occur



EXCLUSIVE OR - Output fault occurs if exactly one of the input faults occurs



PRIORITY AND - Output fault occurs if all of the input faults occur in a specific sequence (the sequence is represented by a CONDITIONING EVENT drawn to the right of the gate)



INHIBIT - Output fault occurs if the (single) input fault occurs in the presence of an enabling condition (the enabling condition is represented by a CONDITIONING EVENT drawn to the right of the gate)

TRANSFER SYMBOLS



TRANSFER IN - Indicates that the tree is developed further at the occurrence of the corresponding TRANSFER OUT (e.g., on another page)



TRANSFER OUT - Indicates that this portion of the tree must be attached at the corresponding TRANSFER IN

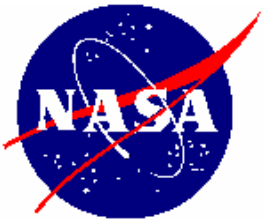
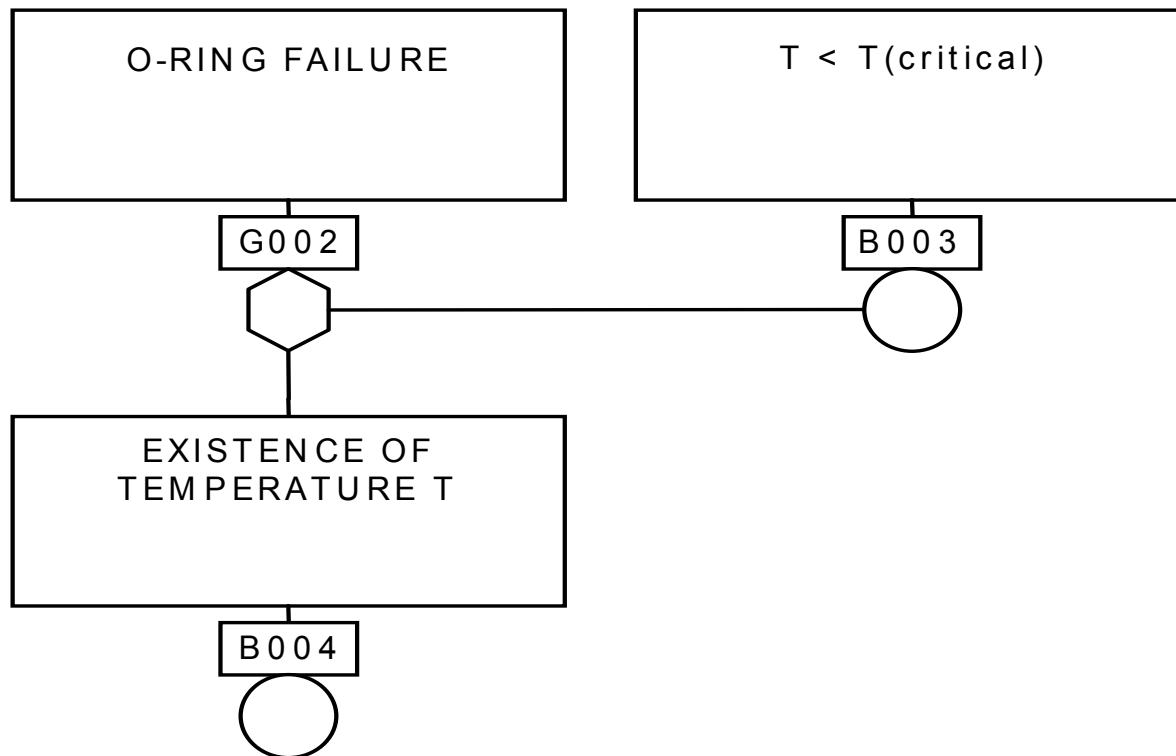
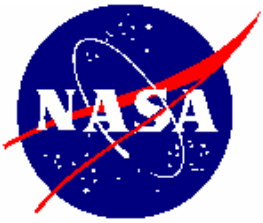
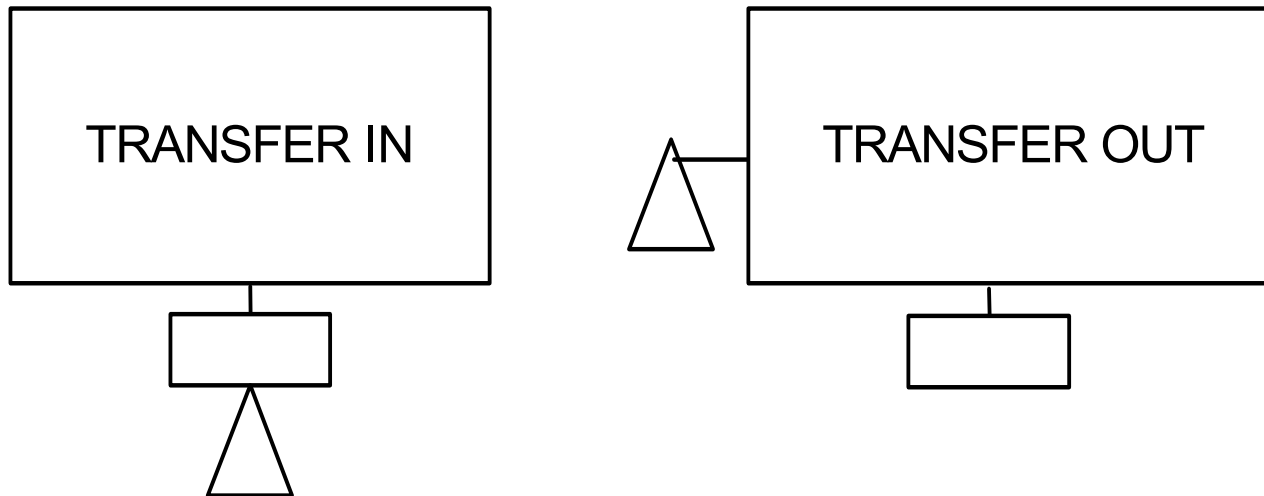


Illustration of the Inhibit Gate





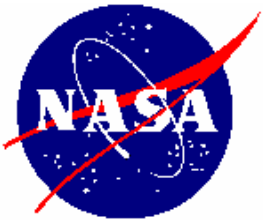
Transfer Gates





Review Questions

1. What is a FT constructed as part of the resolution process?
2. What is the basic paradigm of FTA?
3. Can the top event be a system success?
4. Can any relation be expressed by AND and OR gates?
5. Can the FT be terminated at events more general than basic component failures?
6. Can a FT be developed to a level below a basic component level, e.g. to a piecepart level?
7. Can an intermediate or basic event in the fault tree consist of non-failure of a component?



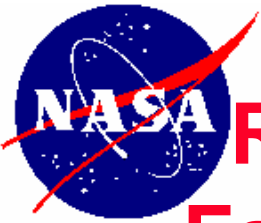
Developing the Fault Tree

- 1. Define the top event as a rectangle**
- 2. Determine the immediate necessary and sufficient events which result in the top event**
- 3. Draw the appropriate gate to describe the logic for the intermediate events resulting in the top event**
- 4. Treat each intermediate event as an intermediate level top event**
- 5. Determine the immediate, necessary and sufficient causes for each intermediate event**
- 6. Determine the appropriate gate and continue the process**



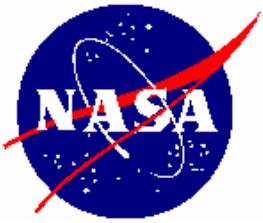
Advise in Developing the Fault Tree

- The system being analyzed for the undesired event needs to be studied and understood before the fault tree is constructed
- If an electrical or hydraulic system is being analyzed, the fault tree is constructed by tracing the causes upstream in the circuit to the basic causes
- For a generalized network or flow, the fault tree is similarly constructed by upstream tracing of the causes



Remember the Four Key Attributes of a Fault Tree

- ✓ **Top Event-** What specific event is being analyzed?
- ✓ **Boundary-** What is inside and outside the analysis?
- ✓ **Resolution-** What are the primary causes to be resolved to?
- ✓ **Initial State-** What is assumed for the initial conditions and states?



Defining the Boundary and Resolution of the Fault Tree

- The *boundary* defines what is *inside* the analysis and what is *outside* the analysis
- The *resolution* defines the *basic causes* to be resolved
- The *boundary* defines the *interfaces* to be included or excluded
- The *resolution* defines what *types of events* are modeled



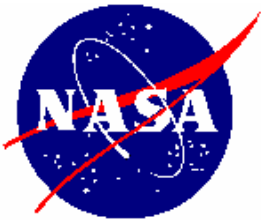
Examples of Boundary Definitions

- **All components shown in a system schematic with detailed system specifications**
- **All major systems identified to comprise an enterprise with detailed system descriptions and their interfaces**
- **The individual steps defined in a process with the detailed process description**
- **The individual processes involved in transforming given inputs into a finished product with detailed descriptions**
- **A software description including coding, flow charts, and detailed descriptions**



Examples of Resolution Definitions

- Resolve basic causes to major components in the system with descriptions of the the included components
- Resolve basic causes to individual tasks in a process with specific listing of the tasks to be included
- Resolve basic causes to major system components, including interfaces among the systems, with detailed descriptions of the components and interfaces
- Resolve the basic causes of software failure to the individual statements in the software program
- Resolve basic causes to major components in the system but do not include interfaces to the system



The Initial State for the Fault Tree

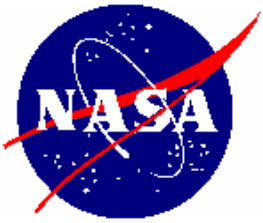
- **The initial state for the FTA defines the initial states of components, initial conditions, and initial inputs assumed**
- **The initial states for the components involve what components are assumed to be initially operational**
- **The initial state can also involve the past history description of the component**
- **Initial conditions include assumed environments and operational conditions**
- **Initial inputs include assumed initial commands, assumed failures existing, and assumed events that have occurred**



A Fault Tree Distinguishes Faults Versus Failures

- The **intermediate events** in a fault tree are called **faults**
- The **basic events, or primary events**, are called **failures** if they represent failures of components
- It is important is to clearly define each event as a fault or failure so it can be further resolved or be identified as a basic cause

Write the statements that are entered in the event boxes as faults; state precisely what the fault is and the conditions under which it occurs. Do not mix successes with faults.



A Fault Tree Distinguishes a Component Fault From System Fault

- **For each event, ask the question whether the fault is a state of component fault or a state of system fault.**
- **The answer determines the type of gate to construct**

If the answer to the question, “Is this fault a component failure?” is “Yes,” classify the event as a “state of component fault.” If the answer is “No,” classify the event as a “state of system fault.”



Component Fault Versus System Fault (Continued)

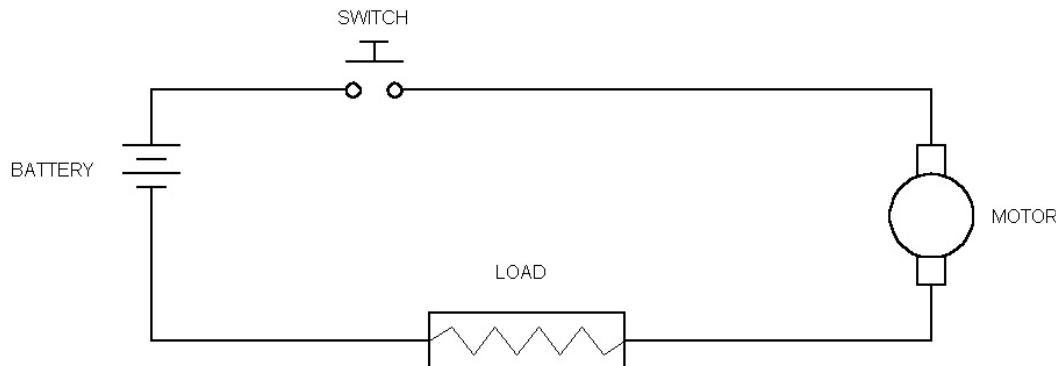
- **For a state of component fault the component has received the proper command**
- **For a state of system fault the proper command may have not been received or an improper command may have been received**
- **The event description needs to clearly define the conditions to differentiate these different faults**



Gates for Component Versus System Faults

- For a state of component fault use an OR gate if the fault is not a failure (basic event)
- For a state of system fault the gate depends on the event description

If the fault event is classified as “state of component,” add an OR-gate below the event and look for primary, secondary and command failure modes. If the fault event is classified as “state of system,” look for the minimum necessary and sufficient immediate cause or causes. A “state of system” fault event may require an AND-gate, an OR-gate, an INHIBIT-gate, or possibly no gate at all. As a general rule, when energy originates from a point outside the component, the event may be classified as “state of system.”



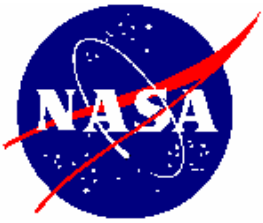
Example of Component Versus System Faults

OPERATING STATE	
FAULT	CLASSIFICATION
Switch fails to close when thumb pressure is applied.	State of component
Switch inadvertently opens when thumb pressure is applied	State of component
Motor fails to start when power is applied to its terminals.	State of component
Motor ceases to run with power applied to terminals	State of component
STANDBY STATE	
FAULT	CLASSIFICATION
Switch inadvertently closes with no thumb pressure applied.	State of component
Motor inadvertently starts.	State of system

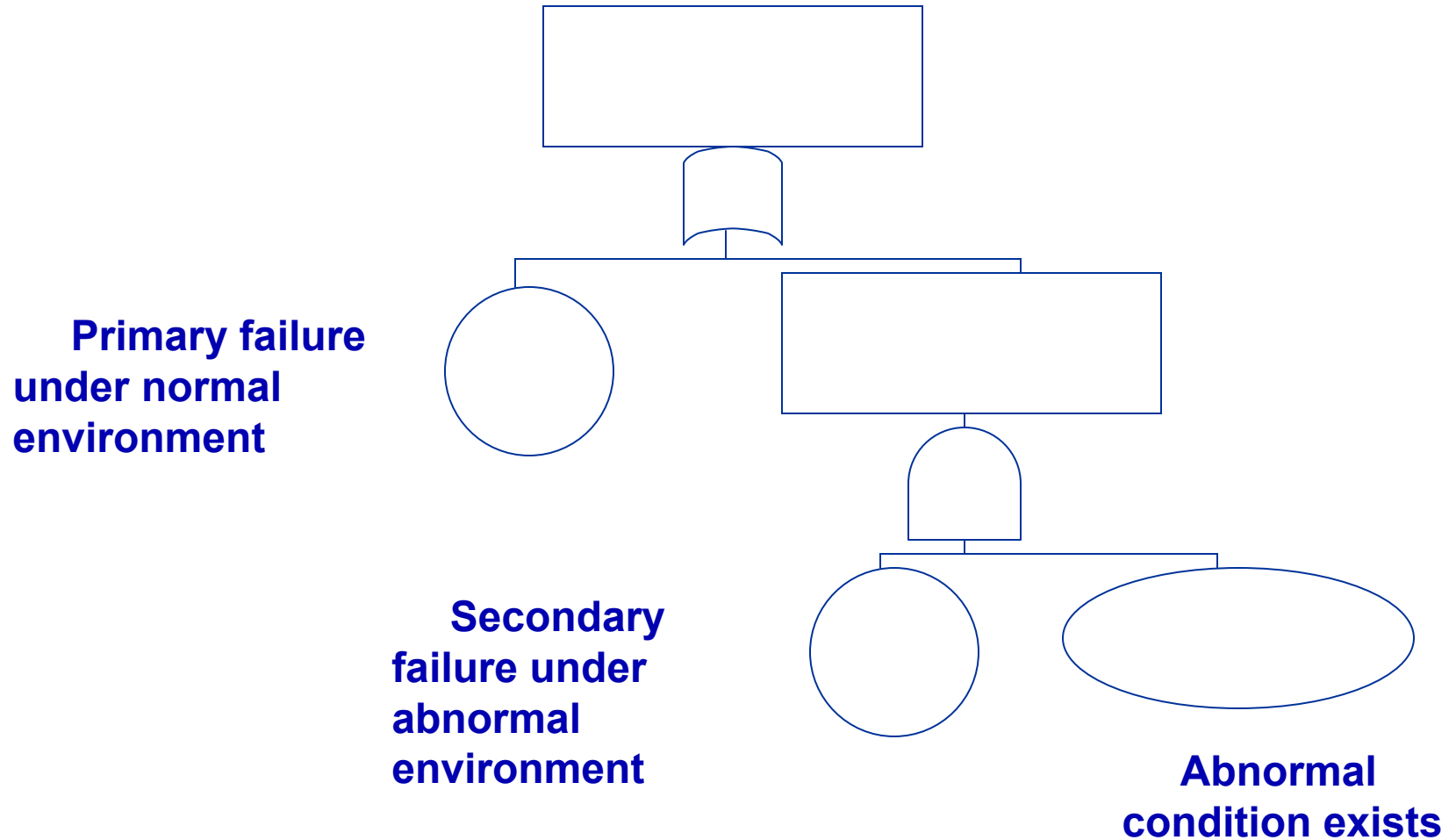


Primary Failure Versus Secondary Failure

- **A failure can be further resolved into a primary failure OR secondary failure**
- **A primary failure is a failure within design environments**
- **A secondary failure is a failure outside design environments**
- **Usually secondary failures are not included unless abnormal conditions are modeled**
- **If secondary failures are included then the secondary failure is resolved into the abnormal condition existing AND the failure occurring**



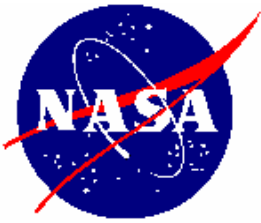
A Primary-Secondary Failure Gate





Secondary Failure Modeling Guidelines

- Include a secondary failure when an abnormal environment is of specific focus
- Include a secondary failure when an abnormal environment can have a non-negligible probability of existing
- Otherwise, as a general rule, do not include secondary failures in the fault tree since they can greatly compound the complexity of the fault tree



The No Miracle Rule

- Do not assume abnormal conditions will occur to prevent a fault from propagating
- In particular, do not assume a failure of another component will occur to prevent a fault from propagating

- If the normal functioning of a component propagates a fault sequence, then
- it is assumed that the component functions normally.



Naming Schemes For the Fault Tree

- Each Gate and Event on the Fault Tree needs to be named
- The Name should ideally identify the Event Fault and the What and When Conditions
- Software packages have default names that can be used but are not descriptive
- Basic events should in particular be named to identify the failure mode
- What is important is that the same event be given the same name if it appears at different locations



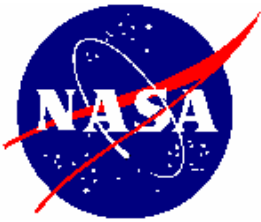
Example of Simple Naming Scheme

Component Type	Component Failure Mode	Description
HX	F	Heat Exchanger Cooling Capability Fails
HX	J	Heat Exchanger Tube Rupture
HX	P	Heat Exchanger Plugs
IN	F	Inverter No Output
IR	F	Regulating Rectifier No Output
IV	F	Static Voltage Regulator No Output
LC	D	Logic Circuit Fails to Generate Signal
LS	D	Level Switch Fails to Respond
LS	H	Level Switch Fails High
LS	L	Level Switch Fails Low



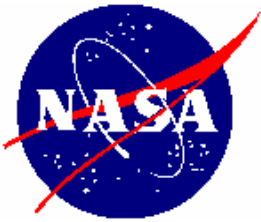
More Complex Naming Schemes

S u b s y s t e m	P R A F a i l u r e D a t a		f a i l u r e
	C o m p o n e n t I D	C o m p o n e n t T y p e	M o d e
LH2	A_O_LH2_DISCVL_FTCM A_O_LH2_DISCVL_FTCE A_O_LO2_DISCVL_FTCE	Valve, 17" Disconnect	fails to close
LDS	E_O_LDS_ACTLUL_JAM E_O_LDS_ACTRUL_JAM E_O_LDS_ACTNUL_JAM	Actuator, hyd uplock	jams



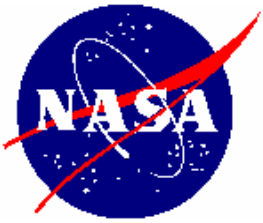
Advise in Defining Ground Rules for an FTA

1. For FT quantification, model to the highest level for which data exists and for which there are no common hardware interfaces
2. Do not generally model wire faults because of their low failure rates
3. Do not generally model piping faults because of their low failure rates
4. Do not further develop an AND gate with three independent inputs if there are lower order contributing combinations
5. Do not further develop an event to an OR gate if there are higher probability input events

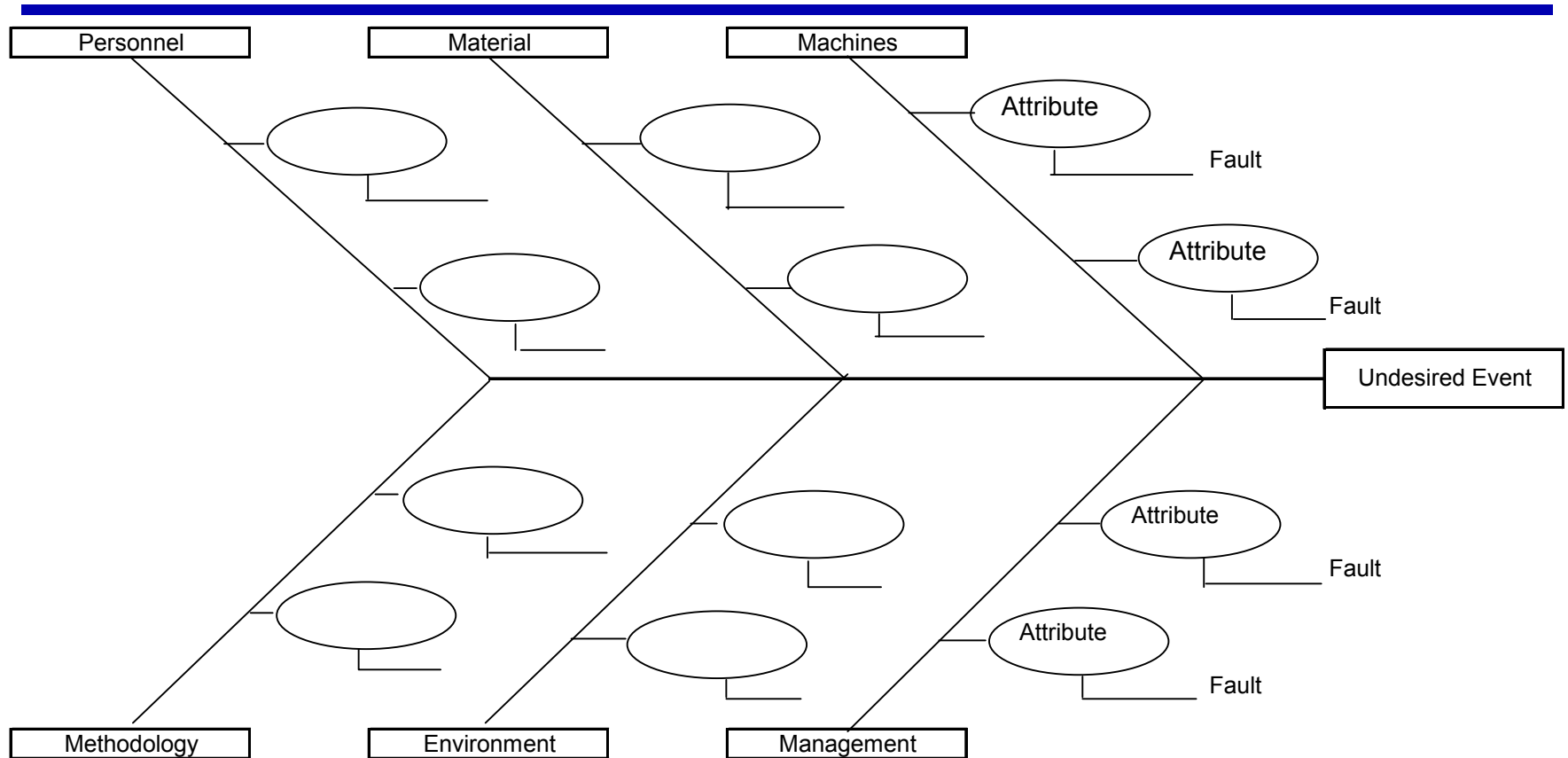


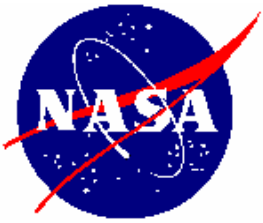
The Fault Tree Versus the Ishikawa Fishbone

- A fault tree is sometimes erroneously thought to be an example of an Ishikawa Fishbone Model
- The fishbone is a loosely-structured, brain-storming tool for listing potential causes of an undesired event
- Fault tree analysis is a stepwise formal process for resolving an undesired event into its immediate causes
- The fault tree displays the stepwise cause resolution using formal logic symbols



The Ishikawa Fishbone Diagram





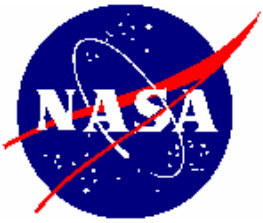
Review Questions

- 1. What is the basic paradigm of FTA?**
- 2. How is FTA different from a Fishbone Model?**
- 3. Can all relations be expressed by AND and OR gates?**
- 4. What are the four key attributes of an FTA?**
- 5. What is the difference between a fault and a failure as defined in FTA? Is this distinction used in other areas?**
- 6. How is a state of component fault modeled?**
- 7. Why can't there be more definite rules for modeling a state of system fault?**



Mono-propellant Propulsion System

- A mono-propellant propulsion system provides an example for FTA
- The system is pressure fed and provides thrust for a vehicle while in orbit
- Additional support systems are not considered
- Different fault trees can be constructed depending on the failure to be modeled

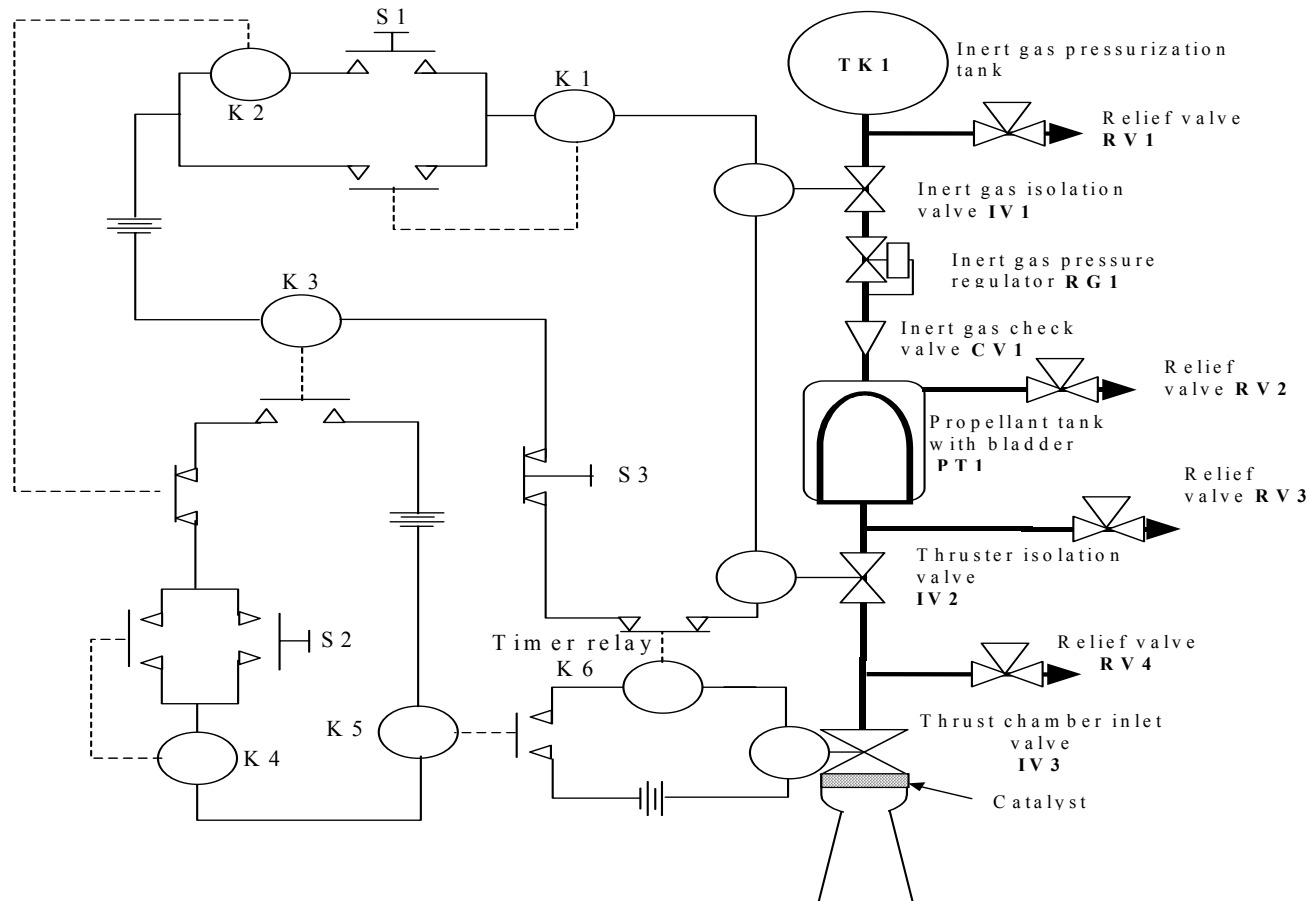


Defining the FT Key Attributes for the Monopropellant System Fault Tree

- ***Top Event*** – Defined based on the specific system failure mode to be analyzed.
- ***Boundary*** – Extracted from the system logic diagrams.
- ***Resolution*** – Include the major components in the system diagram. Do not include wiring faults.
- ***Initial State*** – Dependent on the system failure mode to be analyzed.



System Schematic and Boundaries

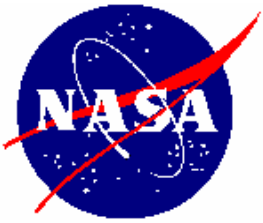


Monopropellant Propulsion System



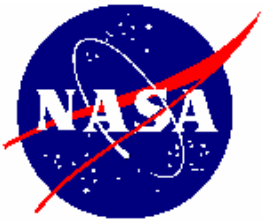
System Components for the FTA

TK1 – Propellant Storage Tank	PT1- Propellant Tank 1
RV1 – Relief Valve 1	K1 – Arming Relay K1
RV2 – Relief Valve 2	K2 – Firing Protection Relay
RV3 – Relief Valve 3	K3 – Arming Relay
RV4 – Relief Valve 4	K4 – Firing Relay
IV1 – Isolation Valve 1	K5 – Firing Relay
IV2 – Isolation Valve 2	K6 – Timing Relay
IV3 – Isolation Valve 3	S1 – Arming Switch
RG1 – Regulator 1	S2 – Firing Switch
CV1 – Check Valve 1	S3 – Emergency Cutoff Switch



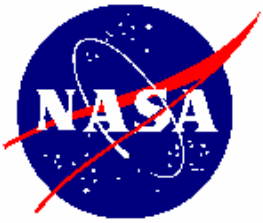
System Description: Basic Operation

The system consists of a reservoir TK1 of inert gas that is fed through an isolation valve IV1 to a pressure regulator RG1. The pressure regulator RG1 senses pressure downstream and opens or closes to control the pressure at a constant level. A check valve, CV1 allows passage of the inert gas to the Propellant Tank PT1. Separating the inert gas from the propellant is a bladder that collapses as propellant is depleted. Propellant is forced through a feed line to the Thruster Isolation Valve IV2 and then to the Thrust Chamber Inlet Valve IV3. For the Thruster to fire, the system must first be armed, by opening IV1 and IV2. After the system is armed, a command is sent to IV3, to open, allowing H_2O_2 into the thrust chamber. As the propellant passes over the catalyst, it decomposes producing the byproducts and heat and the expanding gas that creates the thrust. The relief valves RV1-4 are available to dump propellant overboard should an overpressure condition occur in any part of the system.



System Description: Arming and Thrust

The electrical command system controls the arming and thrusting of the propellant system. To arm, switch S1 is momentarily depressed, allowing electromotive force (emf) to activate relay switches K1, K2 and K3, and open valves IV1 and IV2. K1 closes and sustains the emf through the arming circuit. K2 momentarily opens to preclude the inadvertent firing of the system during the transition to the armed mode, and closes when S1 is released. K3 closes in the firing circuit. The system is now armed with power supplied to sustain IV1 and IV2 in the open position. When firing switch S2 is momentarily depressed, K4 closes sustaining the firing circuit. K5 closes completing the circuit for K6, which begins timing to a predetermined time for the thruster to fire. The completed circuit opens IV3 and thrusting begins.



System Description: Termination of Thrusting

When K6 times out, it momentarily opens breaking the arming circuit and opening K1. Power is removed from the IV1 and IV2 relays and both valves are spring-loaded closed. K3 opens breaking the firing circuit, which opens K4 and K5. IV3 is spring-loaded closed, and the system is in now in the dormant mode. Should K6 fail and remain closed after timing out, the system can be shut down manually by depressing S3, which breaks the arming circuit, opening K1 and closing IV1 and IV2. The firing circuit relay switch K3 will open breaking the firing circuit, which causes K4 and K5 to open. When K5 opens, IV3 will be spring-loaded closed, and the system will be in the dormant mode.

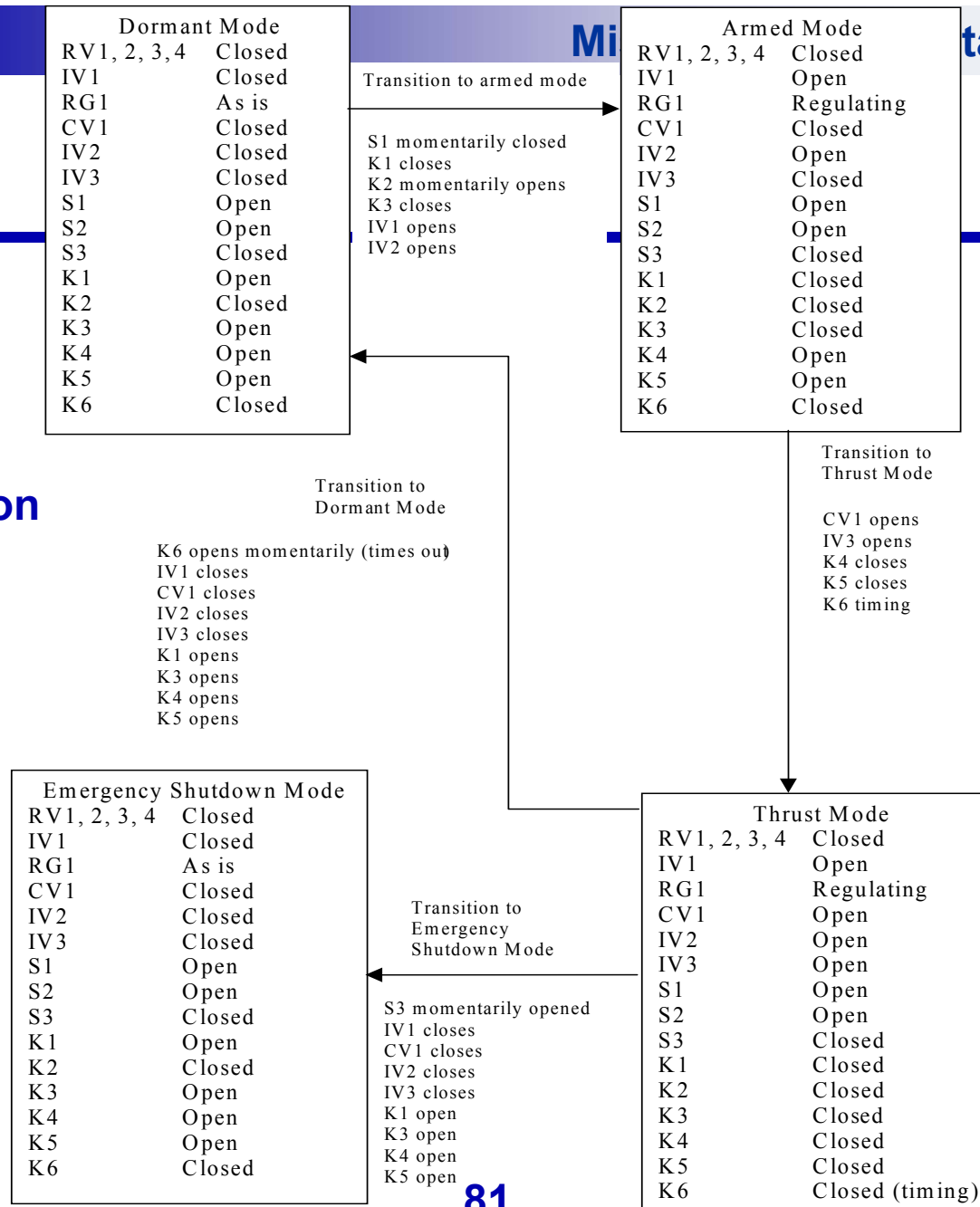


Summary of System Operation

1. **Depress Arming Switch S1. Relays K1 and K3, are energized and close. This results in Isolation Valves IV1 and IV2 opening. Propellant is consequently supplied up to Isolation Valve IV3. Relay K2 briefly opens to preclude inadvertent firing and closes when S1 is released.**
2. **Depress Firing Switch S2. Relays K4 and K5 are energized and close. Isolation Valve IV3 opens and thrusting begins. The closure of K5 initiates the Timing Relay K which times out after a given period opening the relay. The arming circuit is de-energized, closing the Isolation Valves IV1 and IV2 which are spring loaded. Propellant supply stops and the thrusting stops. Manual Switch S3 is a backup emergency.**

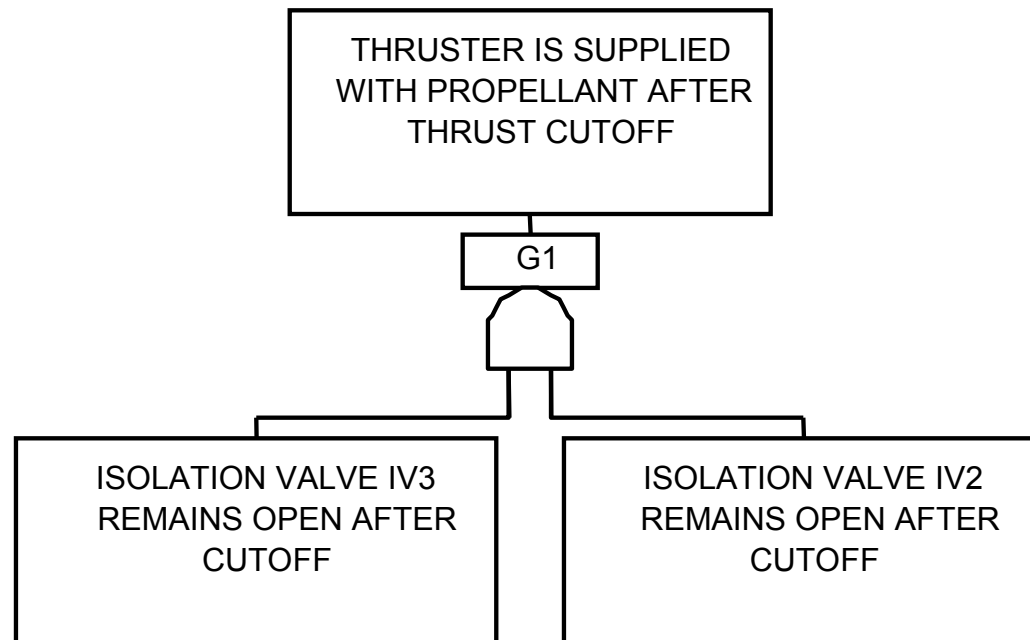


State transition diagram

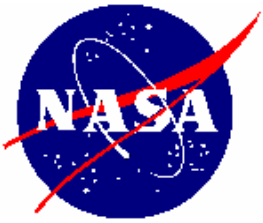




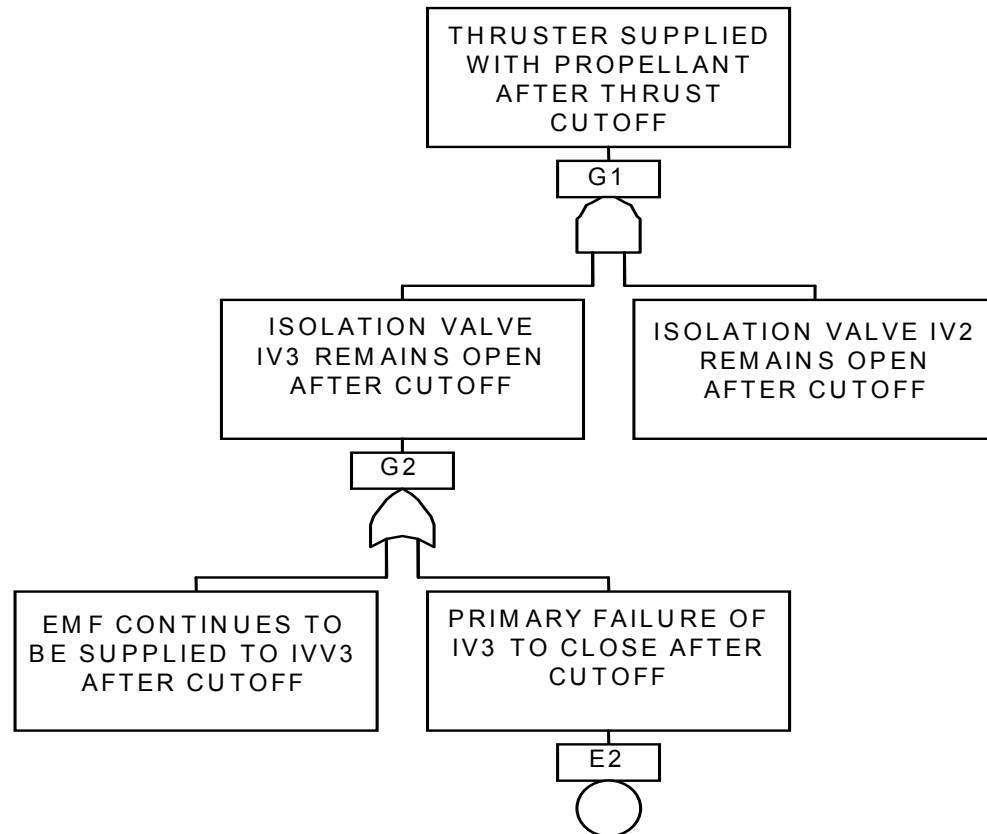
Top Event Structure for “Thruster Supplied with Propellant After Thrust Cutoff”



Fault Tree Construction – Step 1

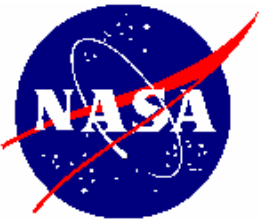


Fault Tree Construction – Step 2

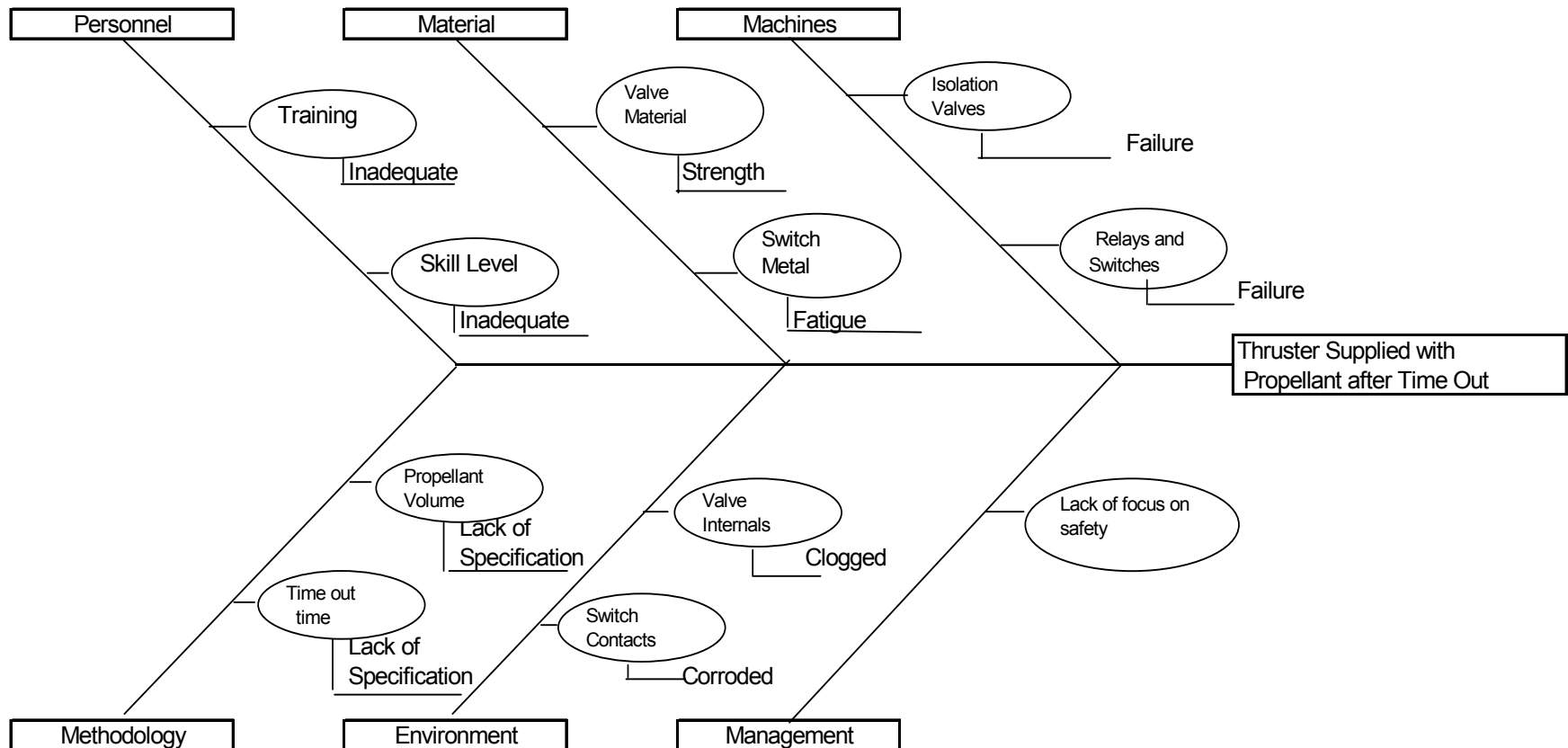




Continue Development of the Fault Tree for the Top Event “Thruster Supplied with Propellant after Thrust Cutoff



Example of Fishbone for the Monopropellant Example





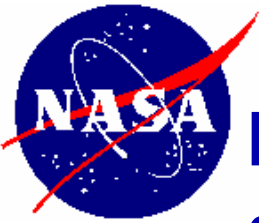
Treatment of Human Errors in FTA

- Human errors are classified into two basic types- errors of omission and errors of commission
- *An error of omission* is not doing a correct action
- *An error of commission* is doing an incorrect action
- Human errors are modeled as basic events in a FT, similarly to component failures
- Human errors need to be considered whenever a human interfaces with the component or system
- The failure modes need to be expanded to include failure induced by the human

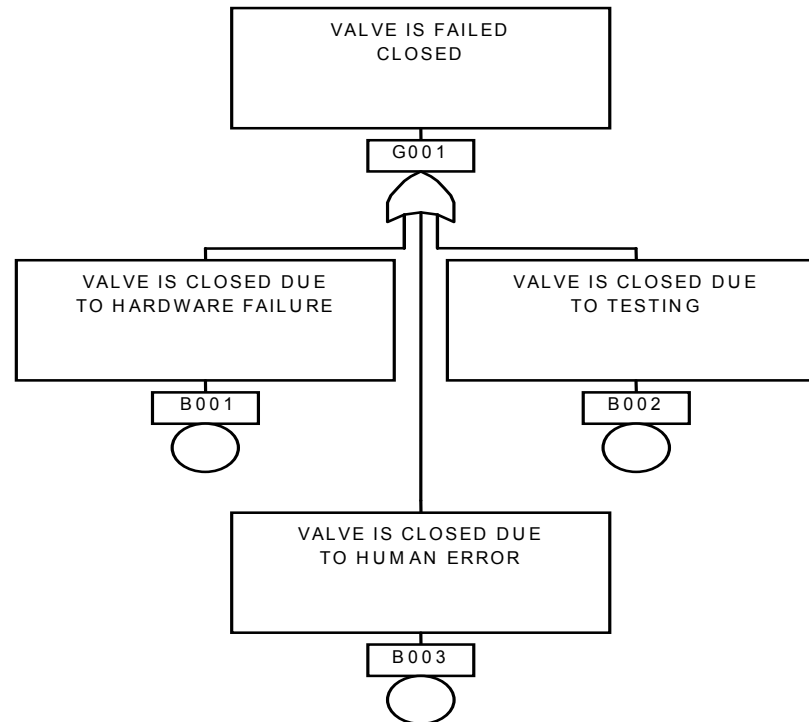


Human Errors Commonly Modeled

- **Test and maintenance related errors**
 - **Errors causing initiating events**
 - **Procedural errors during an incident or accident**
 - **Errors leading to inappropriate actions**
 - **Detection and Recovery errors**

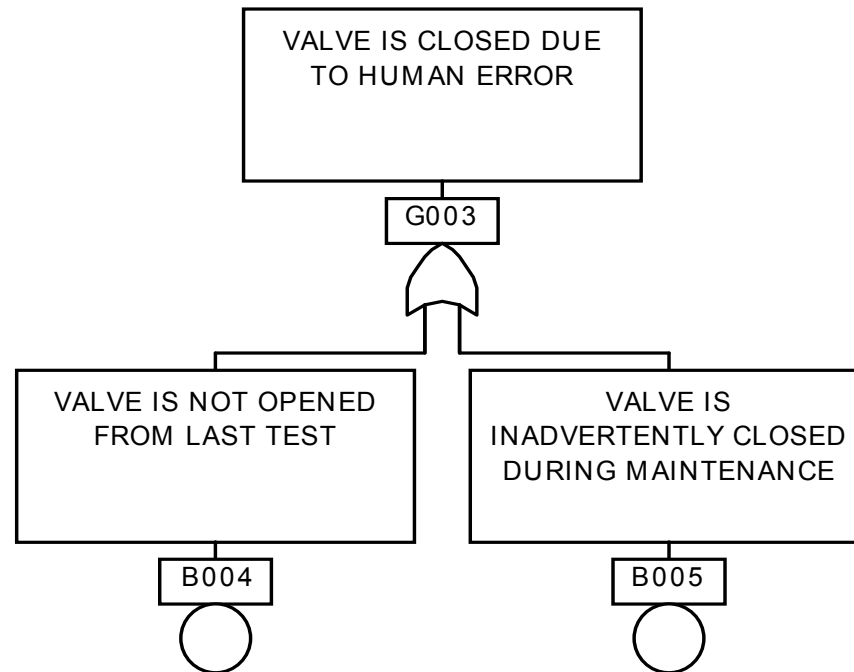


Modeling of Human Error Contribution and Test Contribution





Modeling of More Detailed Human Error Contributions





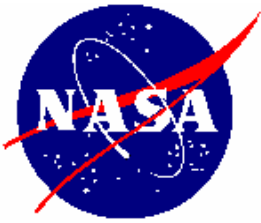
Questions to Determine Whether to Include a Human Error Contribution

- 1. Can an crew error cause the fault?**
- 2. Can a test or maintenance error cause the fault?**
- 3. Can a processing error cause the fault?**
- 4. Can a calibration error cause the fault?**
- 5. Can the fault be recovered by a human action?**
- 6. Is a human action necessary for proper functioning?**
- 7. Can an inadvertent human error result in other faults occurring?**



Examples of Human Error Probabilities

CREAM Nominal Values and Uncertainty Bounds for Cognitive Failures				
Cognitive Function	Generic Failure Type	5% Lower Bound	Median	95% Upper Bound
Observation	Wrong object observed	3.0E-04	1.0E-03	3.0E-03
	Wrong Identification	1.0E-03*	3.0E-03*	9.0E-03*
	Observation Not Made	1.0E-03*	3.0E-03*	9.0E-03*
Interpretation	Faulty diagnosis	9.0E-02	2.0E-01	6.0E-01
	Decision error	1.0E-03	1.0E-02	1.0E-01
	Delayed interpretation	1.0E-03	1.0E-02	1.0E-01
Planning	Priority error	1.0E-03	1.0E-02	1.0E-01
	Inadequate plan	1.0E-03	1.0E-02	1.0E-01
Execution	Action of Wrong Type	1.0E-03	3.0E-03	9.0E-03
	Action at wrong time	1.0E-03	3.0E-03	9.0E-03
	Action on wrong object	5.0E-05	5.0E-04	5.0E-03
	Action out of sequence	1.0E-03	3.0E-03	9.0E-03
	Missed action	2.5E-02	3.0E-02	4.0E-02



Performance Shaping Factors for Human Error Probabilities

CREAM Performance Factors					
Factor	Level	Cognitive Function			
		Observation	Interpretation	Planning	Execution
Adequacy of Organization	Very efficient	1.0	1.0	0.8	0.8
	Efficient	1.0	1.0	1.0	1.0
	Inefficient	1.0	1.0	1.2	1.2
	Deficient	1.0	1.0	2.0	2.0
Working Conditions	Advantageous	0.8	0.8	1.0	0.8
	Compatible	1.0	1.0	1.0	1.0
	Incompatible	2.0	2.0	1.0	2.0
Adequacy of Man Machine Interface	Supportive	0.5	1.0	1.0	0.5
	Adequate	1.0	1.0	1.0	1.0
	Tolerable	1.0	1.0	1.0	1.0
	Inappropriate	5.0	1.0	1.0	5.0
Availability of Procedures	Appropriate	0.8	1.0	0.5	0.8
	Acceptable	1.0	1.0	1.0	1.0
	Inappropriate	2.0	1.0	5.0	2.0
Number of simultaneous Goals	Fewer than capacity	1.0	1.0	1.0	1.0
	Matching capacity	1.0	1.0	1.0	1.0
	More than capacity	2.0	2.0	5.0	2.0
Available Time	Adequate	0.5	0.5	0.5	0.5
	Temporarily inadequate	1.0	1.0	1.0	1.0
	Continuously inadequate	5.0	5.0	5.0	5.0
Time of day	Day-time	1.0	1.0	1.0	1.0
	Night-time	1.2	1.2	1.2	1.2
Adequacy of Training/Preparation	Adequate, high experience	0.8	0.5	0.5	0.8
	Adequate, low experience	1.0	1.0	1.0	1.0
	Inadequate	2.0	5.0	5.0	2.0
Team Collaboration Quality	Very efficient	0.5	0.5	0.5	0.5
	Efficient	1.0	1.0	1.0	1.0
	Inefficient	1.0	1.0	1.0	1.0
	Deficient	2.0	2.0	2.0	5.0



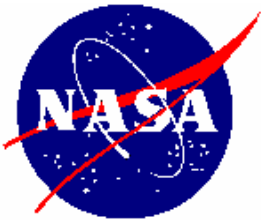
Examples of Human Error Probability Assessments

A_O_ATC_HUMSTBYPMP_FTC	ATCS	Fail to Transfer	Standby Pump	1.00E-03	3	ATC-HRA-
A_O_ATC_HUMHF_PLG	ATCS	Fail to Defrost	HI FES	1.00E-03	3	ATC-HRA-
A_O_ATC_HUMTF_PLG	ATCS	Fail to Defrost	TOP FES	1.00E-03	3	ATC-HRA-
A_O_ATC_HUMHTFRCVR_FOF	ATCS	Fail to Defrost	HI and TOP FES	1.00E-03	3	ATC-HRA-
O_O_ATC_HUMSTBYPMPO1_FTC	ATCS	Fail to Transfer	Standby Pump	1.00E-03	3	ATC-HRA-
O_O_ATC_HUMRADISOO1_FOF	ATCS	Fail to Switch to Auto	Rad Cooling Isolation Valve	3.00E-03	3	ATC-HRA-
O_O_ATC_HUMBPCVCONO1_FOF	ATCS	Fail to Transfer	Standby Controller	1.00E-03	3	ATC-HRA-
O_O_ATC_HUMRADISOBO1_FOF	ATCS	Fail to Switch to Auto	Rad Cooling Isolation Valve	3.00E-03	3	ATC-HRA-
O_O_ATC_HUMBPMANO1_FOF	ATCS	Fail to Open	Bypass valve	1.00E-03	3	ATC-HRA-
O_O_ATC_HUMABPASSO1_FOF	ATCS	Fail to Transfer	Standby Controller	1.00E-03	3	ATC-HRA-
O_O_ATC_HUMACVASSO1_FOF	ATCS	Fail to Transfer	Standby Controller	1.00E-03	3	ATC-HRA-
O_O_ATC_HUMATEMPO1_FOF	ATCS	Fail to Transfer	Standby Controller	1.00E-03	3	ATC-HRA-
O_O_ATC_HUMTFO1_PLG	ATCS	Fail to Defrost	TOP FES	1.00E-03	3	ATC-HRA-
O_O_ATC_HUMHTFRCVRO1_FOF	ATCS	Fail to Defrost	HI and TOP FES	1.00E-03	3	ATC-HRA-
O_O_ATC_HUMBBPASSO1_FOF	ATCS	Fail to Transfer	Standby Controller	1.00E-03	3	ATC-HRA-
O_O_ATC_HUMBCVASSO1_FOF	ATCS	Fail to Transfer	Standby Controller	1.00E-03	3	ATC-HRA-
O_O_ATC_HUMBTEMPO1_FOF	ATCS	Fail to Transfer	Standby Controller	1.00E-03	3	ATC-HRA-
E_O_ATC_HUMSTBYPMPE1_FTC	ATCS	Fail to Transfer	Standby Pump	1.00E-03	3	ATC-HRA-
E_O_ATC_HUMSTBYPMPE2_FTC	ATCS	Fail to Transfer	Standby Pump	1.00E-03	3	ATC-HRA-
E_O_ATC_HUMHFE1_PLG	ATCS	Fail to Defrost	HI FES	0.5	N/A	ATC-HRA-



Common Cause Failures in FTA

- **Common cause failures (CCFs) are multiple failures due to a common cause**
- **A CCF example is multiple valves being failed because of a common maintenance error**
- **CCFs are especially impacting for redundancies of similar components**
- **CCFs need to be considered when there is a common test, common maintenance, common supplier, or common abnormal environment**
- **CCFs need to be considered if not explicitly modeled by the common stressor AND the multiple failures**



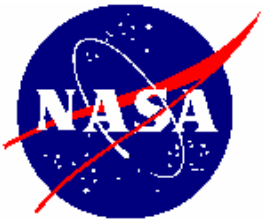
Examples of CCFs

1. A common design or material deficiency that results in multiple components failing to perform a function or to withstand a design environment. Examples include undetected flaws in main engines and low material strengths in turbo pumps.
2. A common installation error that results in multiple components being misaligned or being functionally inoperable. Examples include check valves being installed backwards that remained undetected because they were not tested after installation.
3. A common maintenance error that results in multiple components being misaligned or being functionally inoperable. Examples include multiple valves remaining in a misaligned position after maintenance.
4. A common harsh environment such as vibration, radiation, moisture or contamination that causes multiple components to fail.



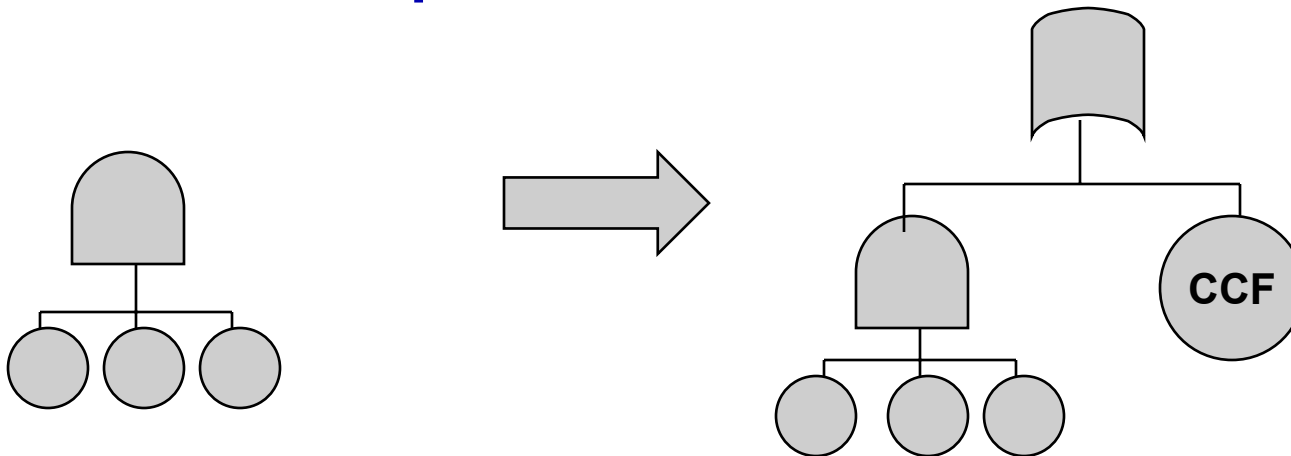
Examples of CCFs Usually Included in FTA

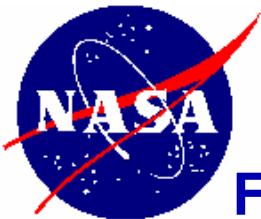
- Redundant sensors having a common calibration procedure
- Redundant components that can be left in the wrong configuration due to a common test or maintenance
- Redundant components that are supplied by the same supplier that have not been independently tested
- Redundant components that have common processing that are not subsequently independently checked



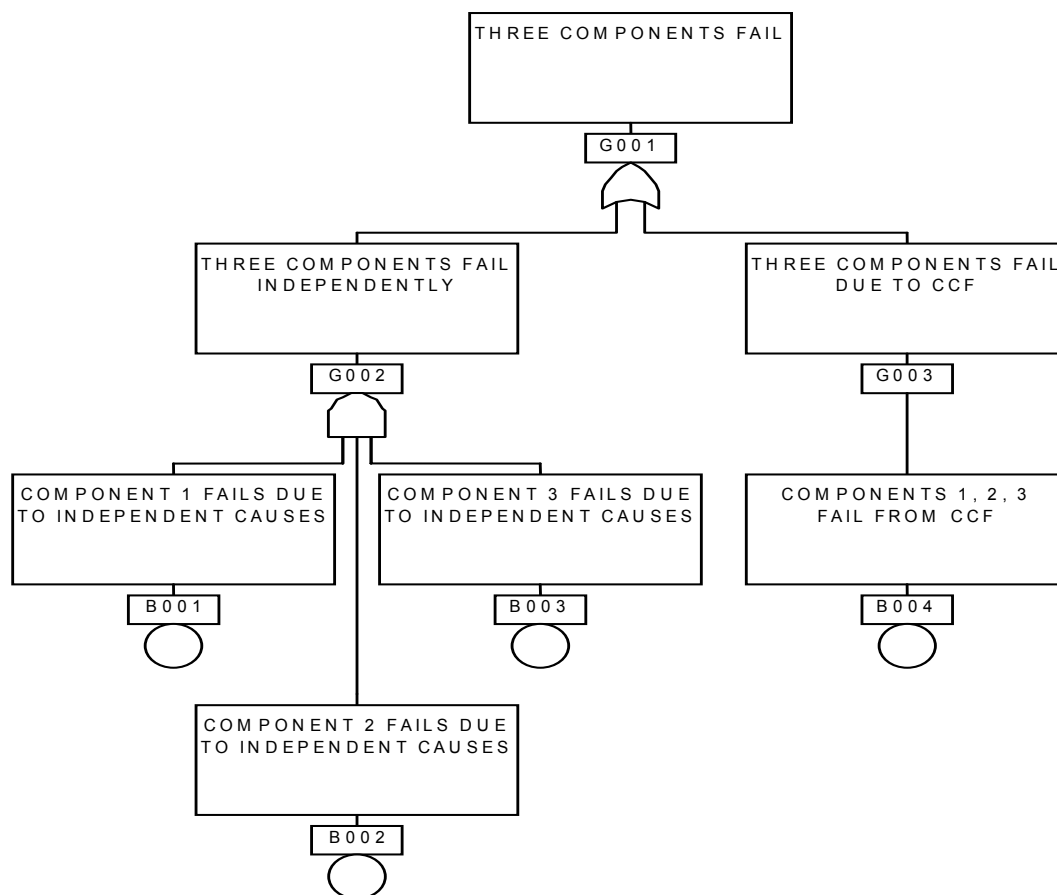
Modeling of CCFs in a FT

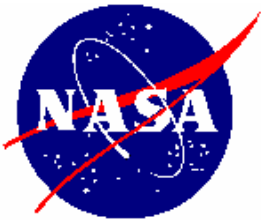
- When considered applicable, a CCF contribution needs to be added to independent failures of similar components
- The AND gate of independent failures is expanded to become an OR gate with the independent failure contribution plus the CCF contribution





Fault Tree Structure including the CCF Contribution





Quantification of CCFs: the Beta Factor Model

β = “beta factor”

= the probability that a failure cause results multiple failures

$$P(C_1 \cdot C_2 \cdot C_3) = P(C_1) \beta$$

β values range from 0.3 to 0.01 when CCF susceptibilities exist

β values are given in various PRAs and data sources, e.g., the Space Shuttle PRA

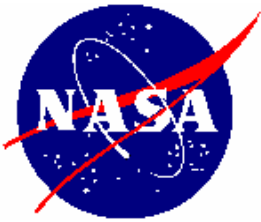


Illustration of the Impact of CCFs

Three redundant components C_1 , C_2 , and C_3

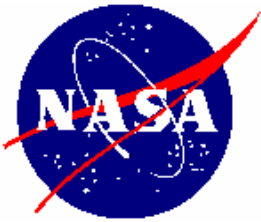
$$P(C) = 1 \times 10^{-3}$$

Independent failure probability:

$$P(C_1 \cdot C_2 \cdot C_3) = 1 \times 10^{-3} 1 \times 10^{-3} 1 \times 10^{-3} = 1 \times 10^{-9}$$

CCF probability ($\beta = 0.01$):

$$P(C_1) \beta = 1 \times 10^{-3} \times 0.01 = 1 \times 10^{-5}$$



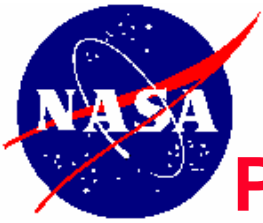
Reviewing the Fault Tree for CCFs

- **AND gates of redundant components are reviewed**
- **A more effective process is to review the minimal cut sets for basic events that can have a single cause**
- **A CCF candidate is a minimal cut set that has:**
 - **Redundant components with a common susceptibility to a single failure cause or single failure enhancing condition**
 - **Multiple human errors that can be committed by a single individual or that have an underlying single procedure**
 - **Multiple components in a common location that can fail due to an external event (e.g., fire or radiation)**
- **Are there any CCF susceptibilities in the Monopropellant System?**



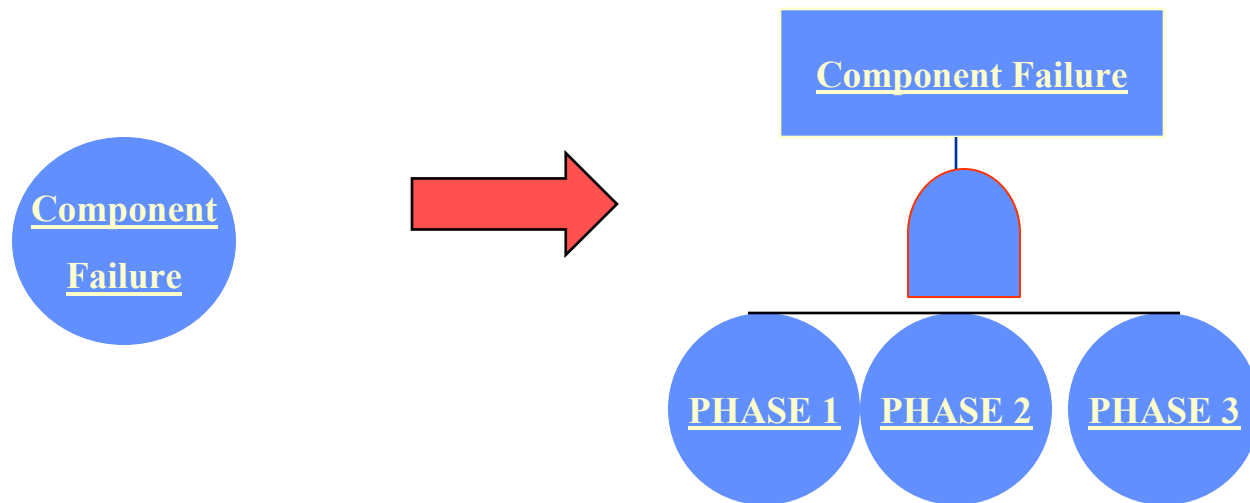
Multiphase FTA

- The system operates in different phases
- The system configuration can change in different phases
- The system success criteria can change
- The basic event probabilities (e.g, component failure rates) can change



Phase Changes in Basic Event Probabilities

- For each phase there are distinct basic event probabilities but no system logic changes
- Each basic event is thus resolved into individual phase events





Phase Changes in Event Probabilities Cont.

- Changes in event probabilities can alternatively be handled in the quantification stage

$$A \cdot B \quad \longrightarrow \quad (A_1 + A_2 + A_3) \cdot (B_1 + B_2 + B_3)$$

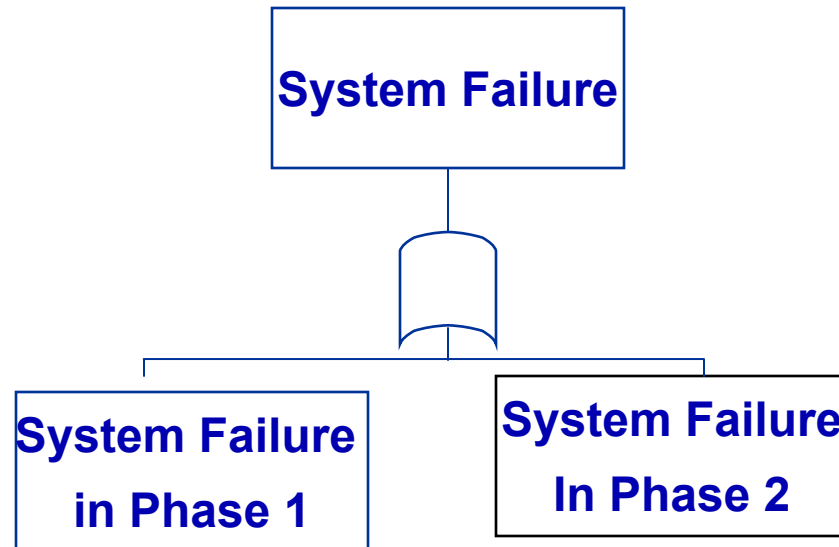
In the above the postscripts 1, 2, and 3 denote the phases.

The formula for the probability of a failure in a phase now includes the probability of non-failure in previous phases



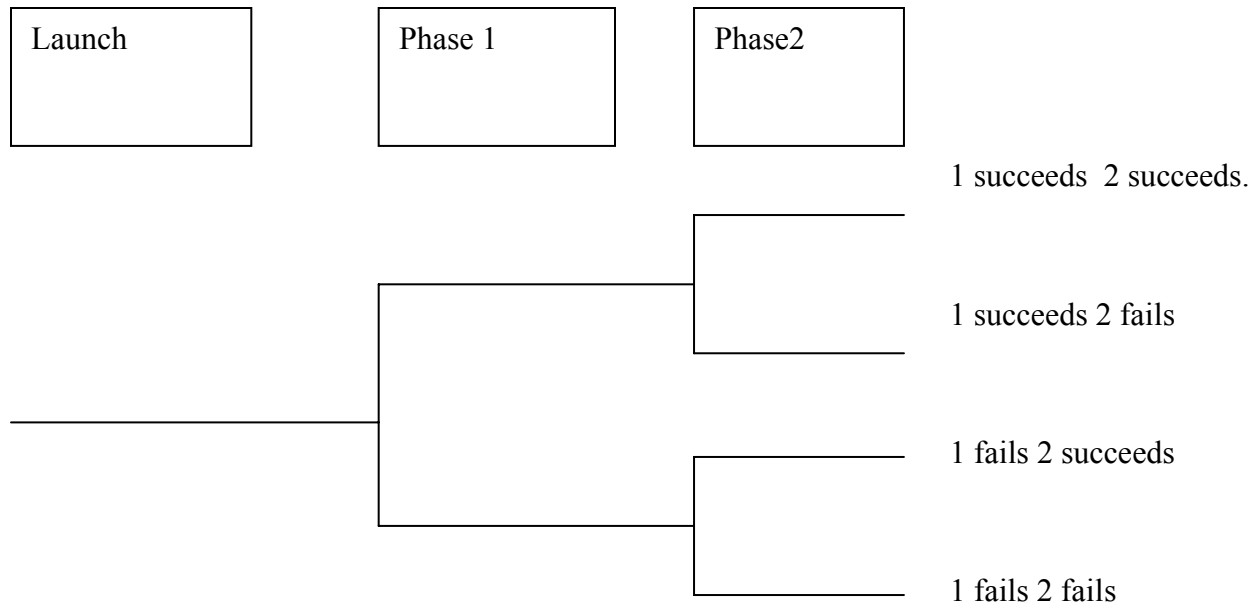
Phase Changes in Logic

- Resolve the System Failure into a Fault Tree for each phase:



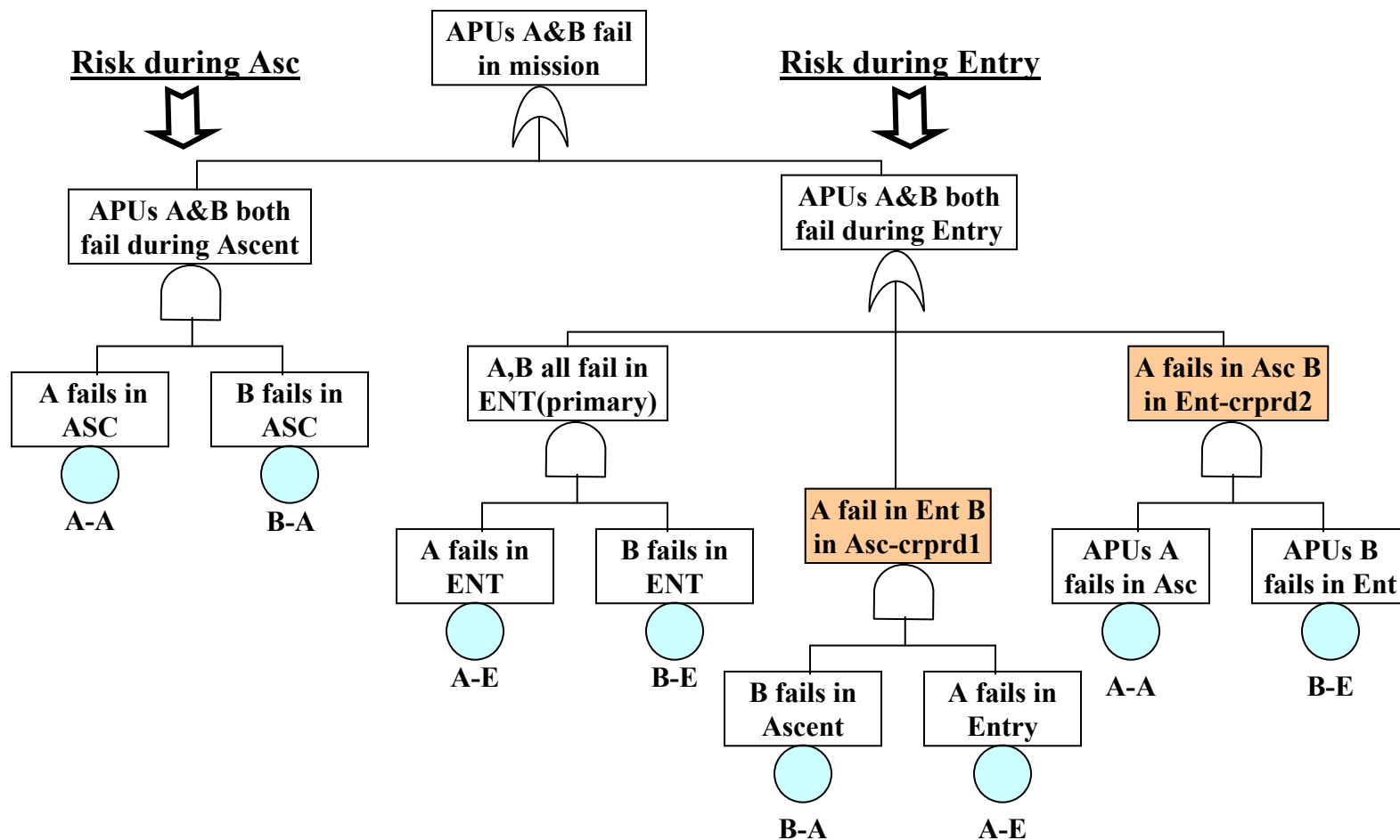


Fault Trees Used in Multi-Phase Event Sequence Modeling





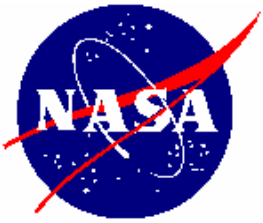
Multi-Phase Fault Tree Modeling Used in the Shuttle PRA





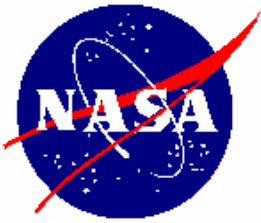
Review Questions

1. What is the difference between CCFs and multiple failures modeled as having a single cause?
2. Do CCFs need to be considered if the Fault Tree is not quantified?
3. Do human errors need to be considered if the Fault Tree is constructed for a system design only?
4. Can the same fault tree be used for multi-phases if the system configuration does not change?
5. Can time-dependencies be handled in the same manner as multi-phases?



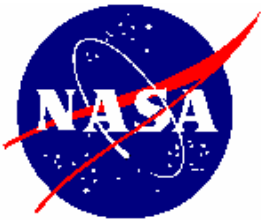
FT Exercise Problem

- **Consider again the Monopropellant System**
- **Construct the FT for the undesired event:**
 - **No propellant supplied to the thruster when the arming command is initiated**
- **Use the same system boundary and resolution as used for the fault tree for the undesired event: thruster supplied with propellant after thrust cutoff**



Evaluating the Fault Tree

- **Constructing the fault tree provides understanding of the system failure logic**
- **The fault tree itself provides a descriptive tool for communication**
- **The fault tree can also be evaluated to obtain critical qualitative and quantitative information**
- **To evaluate the fault tree, the fault tree has to be transformed to an equivalent set of logic equations**



Steps in Qualitatively and Quantitatively Evaluating the Fault Tree

- **Each gate event is expressed as a logic equation of input events**
- **By successive substitution, each gate event is express in terms of basic events**
- **The resulting gate equation is expanded and simplified to be a sum of products (sop)**
- **The resulting equations provide a basis for qualitative and quantitative evaluations**



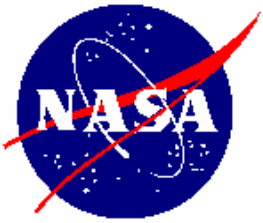
Representation of the Gate Events of the Monopropellant Fault Tree

- G1 – Thruster supplied with propellant after thrust cutoff
- G2 – Isolation valve IV3 remains open after cutoff
- G3 – Isolation valve IV2 remains open after cutoff
- G4 – emf continues to be supplied to IV3 after cutoff
- G5 – emf continues to be supplied to IV2 after cutoff
- G6 – emf continues to be supplied to K5 after cutoff
- G7 – emf continues to be supplied to K3 after cutoff
- G8 – Emergency switch S3 fails to open after cutoff
- G9 - Primary failure of K6 to open after cutoff



Representation of the Basic Events of the Monopropellant Fault Tree

- E1 = Primary failure of IV2 to close after cutoff
- E2 = Primary failure of IV3 to close after cutoff
- E3 = Primary failure of K5 relay to open when emf is removed
- E4 = Primary failure of K3 to open after cutoff
- E5 = Primary failure of K6 to open after timing out
- E6 = Primary failure of K6 timer to time out
- E7 = Operational failure of S3 to open when commanded
- E8 = Primary failure of S3 to open when commanded



Logic Equations for the Monopropellant Fault Tree

$$G1 = G2 \bullet G3$$

$$G2 = G4 + E2$$

$$G3 = G5 + E1$$

$$G4 = G6 + E3$$

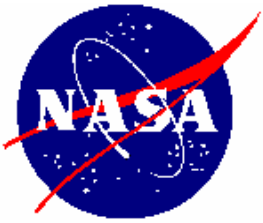
$$G5 = G8 \bullet G9$$

$$G6 = G7 + E4$$

$$G7 = G8 \bullet G9$$

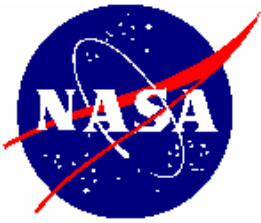
$$G8 = E7 + E8$$

$$G9 = E5 + E6$$



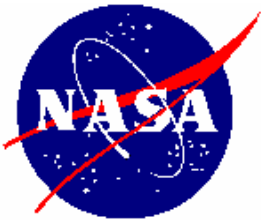
The Minimal Cutsets of a Fault Tree

- A **minimal cutset** (mcs) is a **smallest** combination of primary events, or basic events, causing the top event
- All the primary events need to occur to cause the top event
- Each mcs is thus a causal-combination, i.e., a combination of primary events
- The set of mcs directly link the top event to the primary events
- The complete set of mcs provides the complete set of causes of the top event



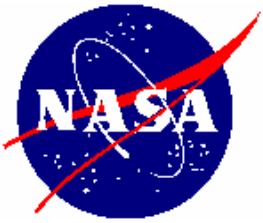
Expanding the Top Event to Obtain the Minimal Cut Sets

1. The fault tree is represented as a set of logic equations
2. Substitution is carried out until the top event is represented entirely in terms of basic events
3. The top event equation is then expanded and simplified to obtain a 'sum of products'
4. In expanding the top event equation, the Boolean distributive law and the law of absorption are used.
5. Each product in the sum of products is then a minimal cut set of the top event



The Minimal Cutsets Provide Key Qualitative Information

- The minimal cutsets directly link the top event to the primary events, or basic events
- The minimal cutset (mcs) *size* is a qualitative ranking of the causal-combination
- A *single element* mcs identifies a single cause of the top event
- The component *types* in the mcs also provides a qualitative ranking of the causal combination
- *Redundant components in a mcs* can be susceptible to a common triggering cause



Basic Boolean Relationships Used in Fault Tree Evaluations

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

Distributive Law

$$A + A = A$$

Identity Union Law

(Identity Absorption Law)

$$A + A \cdot B = A$$

Subset Absorption Law

$$A \cdot A = A$$

Identity Intersection Law

(Idempotent Law)

$$(A + B)' = A' \cdot B'$$

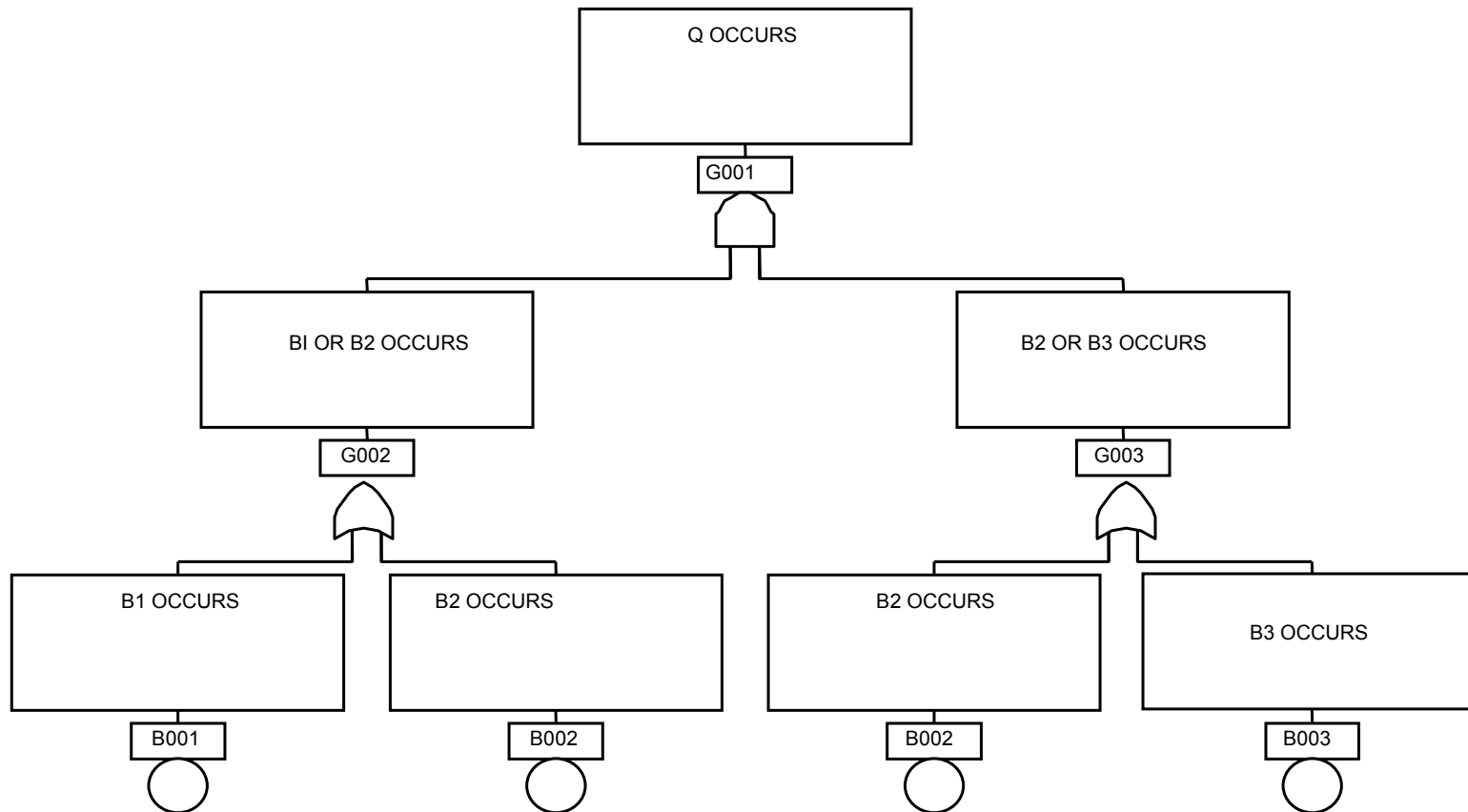
Union Complementation Law

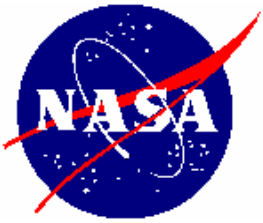
$$(A \cdot B)' = A' + B'$$

Intersection Complementation

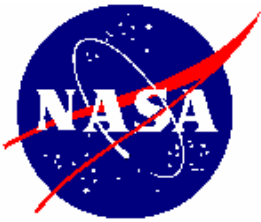


Sample Fault Tree for Boolean Analysis





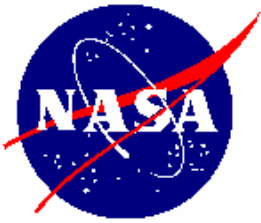
**Problem: Determine the Minimal
Cutsets of the Sample Fault Tree**



The Minimal Cut Set Equation (Sum of Products) for the Monopropellant Tree

Applying the Distributive Law and Laws of Absorption to the Top Event Equation in terms of the Basic Events:

$$G1 = E6 \cdot E7 + E6 \cdot E8 + E5 \cdot E7 + E5 \cdot E8 + E1 \cdot E3 + E1 \cdot E4 + E1 \cdot E2$$



Description of the Minimal Cutsets of the Monopropellant Tree

- E6 • E7=Primary Time out Failure of K6 • Operational Fail to Open of S3
- E6 • E8= Primary Time out Failure of K6 • Primary Fail to Open of S3
- E5 • E7=Primary Fail to Open of K6 • Operational Fail to Open of S3
- E5 • E8= Primary Fail to Open of K6 • Primary Fail to Open of S3
- E1 • E3= Primary Fail to Close of IV2 • Primary Fail to Open of K5
- E1 • E4=Primary Fail to Close of IV2 • Primary Fail to Open of K3
- E1 • E2=Primary Fail to Close of IV2 • Primary Fail to Close of IV3



Review Questions

- 1. Why are the minimal cutsets important?**
- 2. How can the minimal cutsets be obtained for any of the intermediate faults of the fault tree?**
- 3. Why are the minimal cutsets ordered by their size?**
- 4. How can the minimal cutsets be used to check given design criteria, such as having no single failure cause?**
- 5. What can be concluded from the minimal cutsets of the monopropellant fault tree?**



Minimal Cutset Quantification Formulas

$P(T) = P(M_1 + M_2 + \dots + M_N)$ where “+” = Logical OR

$P(T) = \sum P(M_k)$ Sum of Minimal Cutset Probabilities (Rare Event Approximation)

$P(M) = P(E_1)P(E_2)\dots P(E_M)$ Product of Independent Basic Event Probabilities

T = top event

M = minimal cutset

E = basic event



Basic Formulas for Primary Event Probabilities (P(E))

Failure probability for a non-repairable component (or event)

$$P = 1 - \exp(-\lambda T) \sim \lambda T \quad \lambda = \text{component failure rate}$$

T = exposure time

Failure probability for a repairable component

$$P = \lambda \tau / (1 + \lambda \tau) \sim \lambda \tau = \text{component failure rate or event rate}$$

τ = repair time

Constant failure probability for a component

$$P = c \quad c = \text{constant probability (e.g., "per demand")}$$



Details of Formulas: $P=1-\exp(-\lambda T) \sim \lambda T$

λ is the constant component failure rate, e.g., no aging, which is used as a first order approximation. For extreme time dependency, Weibull, etc., can be used

λ depends on the failure mode and environment

For an operating (standby) component λ is the operating (standby) failure rate

The approximation shown above is valid to two significant figures for failure probabilities less than 0.1

The failure exposure time T is the time during which the failure can occur and result in a higher fault

Software packages compute the exact formula



Details of Formulas: $P = \lambda\tau/(1 + \lambda\tau) \sim \lambda\tau$

τ is the average detection plus repair time for the failure

τ depends on the detection and repair process

The above formula is a steady state formula which is generally applicable for times significantly greater than τ

Since $\lambda\tau$ is generally much smaller than one, the above approximation is generally valid to two significant figures

Software packages calculate the exact formula



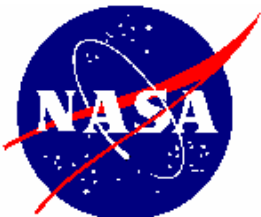
Details of Formulas: $P = c$

The constant probability model is used when applicable probabilities are available

The constant probability model is used when c is the probability per demand, which is called a demand failure rate

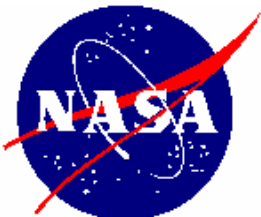
Demand failure rates apply to components starting or changing state, .e.g, relays, circuit breakers, engines starting

Human error rates are expressed as a probability c of human error per action



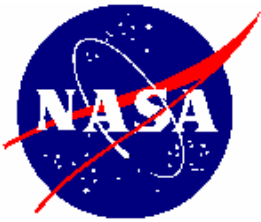
Examples of Component Failure Rates

PRA Failure	failure	Generic failure mode	Failure Rate	Generic Failure Rate	
Component Type	Mode		units	median	mean
Assembly, Gyro Rate	no output	no output	per hour	4.E-06	1.E-05
Assembly, star tracker	no output	no output	per hour	1.E-06	3.E-06
Body Flap	sticking	sticks	per hour	1.E-06	3.E-06
Body Flap	structural failure	structural failure	per hour	1.E-07	3.E-07
Brake	fails to close	fail to close	per demand	1.E-05	3.E-05



Additional Examples of Component Failure Rates

PRA Failure	failure	Generic failure mode	Failure Rate	Generic Failure Rate	
Component Type	Mode		units	median	mean
Pump, hyd	fails to run	fail to operate	per hour	1.E-06	3.E-06
Pump, hyd	fails to start	fail to start	per demand	2.E-04	3.E-04
Sensor, temp	fails hi	fail high	per hour	4.E-07	1.E-06
Sensor, temp	fails low	fail low	per hour	4.E-07	1.E-06
Valve, bypass pneu	fails to open	fail to open	per demand	2.E-04	3.E-04
Valve, bypass pneu	transfers open	transfer open	per hour	1.E-06	3.E-06



Steps in Quantifying Component Failure Probabilities

- 1. Identify the specific component failure mode**
- 2. Determine whether the failure is time-related or demand-related**
- 3. Determine the environment e.g., ground or air**
- 4. Select the appropriate failure rate value**
- 5. For a time-related failure determine the exposure time**
- 6. For a time-related failure, if the failure is repairable determine the repair time**
- 7. For a demand-related failure, determine the number of demands if greater than 1**
- 8. Input into the software package or if a manual evaluation use the appropriate formula to quantify**



Monopropellant Component Failure Data

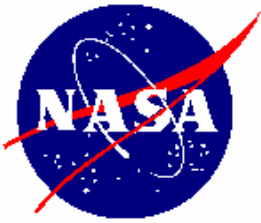
Basic Event	Component Type	Fault Tree Symbols	Failure Mode	Failure Probability
IV	Isolation Valve	E1 E2	Failure to close when EMF is removed	2 E-04
K	Relay Switch Contacts	E3 E4 E5	Failure to return when EMF is removed	3 E-03
K6	Timer Relay	E6	Failure to time out	2 E-02
S	Manual Switch	E7	Operational failure to open Switch	1 E-02
S	Manual Switch	E8	Failure of Switch to open when operated	5 E-05



Quantification of the Minimal Cutsets for the Monopropellant Tree

E6 • E7=Primary Time out Failure of K6•Operational Fail to Open of S3 = $2-02*1-02=2-04$
E6 • E8= Primary Time out Failure of K6•Primary Fail to Open of S3 = $2-02*5-05= 1-06$
E5 • E7=Primary Fail to Open of K6•Operational Fail to Open of S3 = $3-03*1-02= 3-05$
E5 • E8= Primary Fail to Open of K6•Primary Fail to Open of S3 = $3-03*5-05= 1.5-07$
E1 • E3= Primary Fail to Close of IV2•Primary Fail to Open of K5 = $2-04*3-03= 6-07$
E1 • E4=Primary Fail to Close of IV2•Primary Fail to Open of K3 = $2-04*3-03= 6-07$
E1 • E2=Primary Fail to Close of IV2•Primary Fail to Close of IV3 = $2-04*2-04= 4-08$

$$G1=2-04+3-05+1-06+6-07+6-07+1.5-07+4-08 = 2.3-04$$



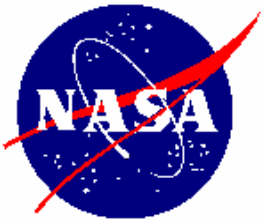
Interpretations of Quantitative Results

- **Basic event probabilities used for quantification generally have large uncertainties**
- **Thus, the quantified probability for the top event and other results generally have large uncertainties**
- **Quantitative results should therefore generally be interpreted as showing the general range of the value, e.g., the order of magnitude**
- **Uncertainty evaluations are carried out to explicitly show the associated uncertainty ranges**
- **Relative contributions and importances obtained from the fault tree generally have smaller uncertainties**



Using Generic Failure Data

- Data bases provide *generic* failure data collected from a variety of sources
- This generic data needs to be screened for the applicable failure mode and environment
- Operational factors or environmental factors are given to scale reference failure data
- The generic data can also be updated using mission specific data
- Bayesian statistical approaches are used in this updating to appropriately handle the information



Using Expert Opinion

- For a variety of basic events, applicable data are not available
- Expert opinion and engineering judgment need thus to be used to estimate the basic event data
- The basis for the estimates need to be documented
- A sufficient range needs to be included with each estimate to cover uncertainties
- Sensitivity studies can be carried out to check the impact of the estimates
- Structured expert-elicitation approaches can be used to increase the fidelity of the estimates



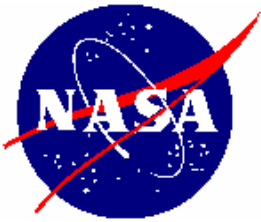
Review Questions

1. Can the sum of products quantification rule for the top event be used for intermediate faults?
2. How is the failure exposure time changed for a component tested or not tested before a launch?
3. How can a constant failure rate model be used to approximate phases or time-dependencies?
4. How can quantification rules for a fault tree be codified to obtain consistent results?
5. How can the quantitative results be used to check the fault tree?



Three Basic Importance Measures Used for Prioritization in FTA

- **FV Importance** (Contribution Importance)- the relative contribution to the top event probability from an event.
- **Risk Achievement Worth RAW** (Increase Sensitivity, Birnbaum Importance)- the increase in the top event probability when an event is given to occur (*probability set to 1*).
- **Risk Reduction Worth RRW** (Reduction Sensitivity)- the reduction in the probability of the top event when an event is given to not occur (*probability set to 0*).



Calculation of the Importance Measures

FV Importance =
$$\frac{\text{Sum of min cut cuts containing the event}}{\text{Sum of all min cut sets}}$$

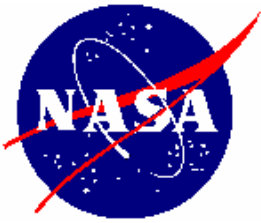
RAW = Top event probability with event probability set to unity
- Top event probability

RRW = Top event probability
- Top event probability with event probability set to zero



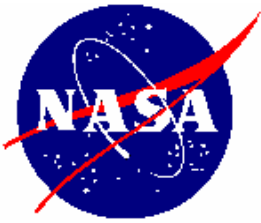
Basic Event Importance Measures for the Monopropellant Example

Basic Event	FV Importance	RRW (Reduction)	RAW (Increase)
Operational Fail to Open S3	0.993	0.993	0.023
Primary Time Out Failure of K6	0.867	0.867	0.01
Primary Fail to Open of K6	0.13	0.13	0.01
Primary Fail to Open of S3	0.005	0.005	0.023
Primary Fail to Close of IV2	0.003	0.003	0.003
Primary Fail to Open of K3	0.003	0.003	0.0002
Primary Fail to Close of IV3	0.0001	0.0001	0.0002



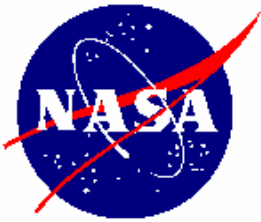
Questions on the Monopropellant Illustration

- 1. Why is the Operational Failure of S3 so high?**
- 2. Why is the Primary Failure of K6 so high?**
- 3. Why is importance of IV2 higher than IV3?**
- 4. What components should be a focus of upgrades?**
- 5. What is the potential improvement from such upgrades?**
- 6. What components can be the focus for relaxations?**
- 7. If the system fails, where should diagnosis be focused?**
- 8. What possible changes can reduce the failure probability?**
- 9. What are other system failures (top events) that can be analyzed?**



Types of Uncertainty in FTA

- **Two types of uncertainty**
 - Modeling uncertainty
 - Parameter uncertainty
- **Modeling uncertainty**
 - Success and failure criteria assumed
 - Contributions excluded
 - Independence assumptions
- **Parameter uncertainty**
 - Uncertainties in data values



Uncertainty Analyses in FTA

- Modeling uncertainties are handled by listing them and carrying out sensitivity analyses
- Parameter uncertainties are handled by using a probability distribution for each data value
 - Median value
 - Mean value
 - 5% and 95% Bounds
 - Type of Distribution (e.g., Beta, Gamma, Lognormal)



FT Uncertainty Propagation

- **Probability distributions are assigned for each basic event data value**
- **Data values having the same estimate are identified as being coupled**
- **The probability distributions are then propagated using Monte Carlo simulations**
- **The probability distribution and associated characteristics are determined for the top event**
 - **Median value**
 - **Mean value**
 - **5% and 95% Bounds**



Validating an FTA

1. Select lower order minimal cutsets and validate that they are failure paths
2. Obtain the minimal cutsets for an intermediate fault and validate selections as failure paths
3. Obtain the success paths and validate selections as true success paths
4. Review failure records and hazard reports to check the coverage of the fault tree
5. Carry out sanity checks on the importance results and probability results



Termination of a Fault Tree Revisited

- Basic events that are resolved
- AND gates with multiple, diverse independent inputs (e.g. 4) when there are smaller failure combinations and with no CCF contribution
- Input events to an OR gate of low probability compared to other inputs
- Intermediate events with upper bound screening probabilities that are determined to have small contributions

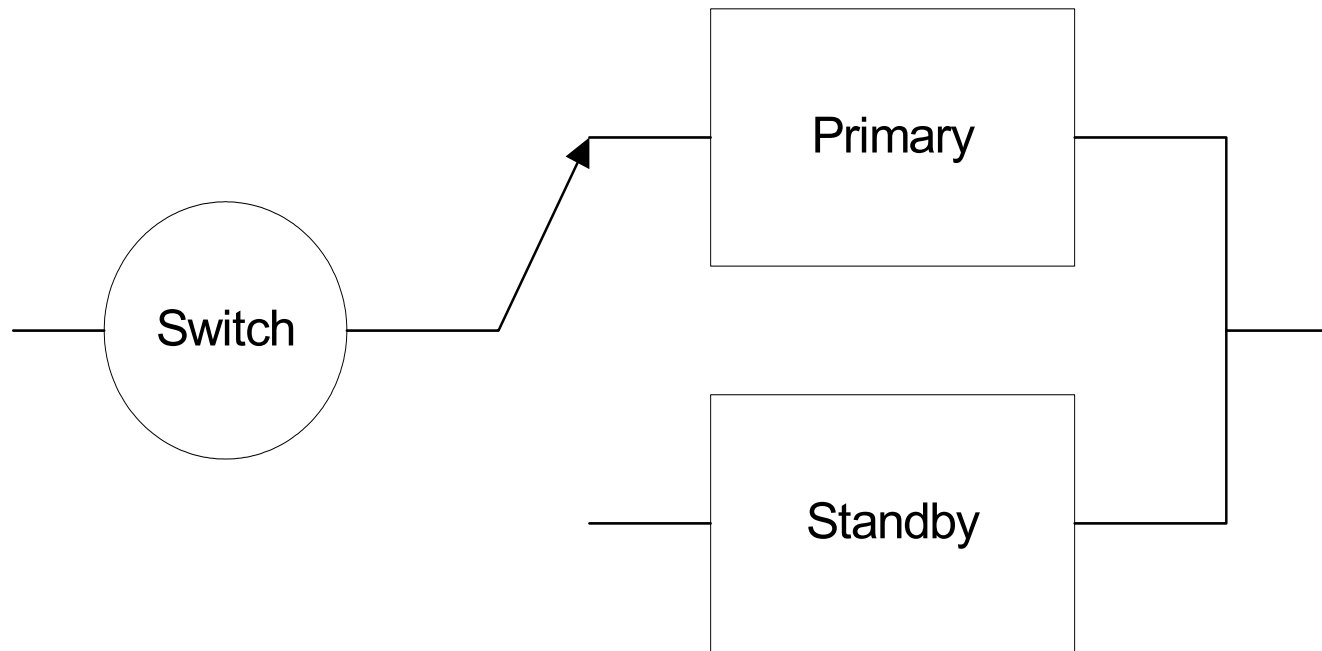


Dynamic Fault Tree Analysis (DFTA)

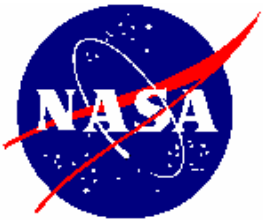
- **DFTA is a term used to refer to analysis of a system which dynamically responds to a failure or a stimulus**
 - **A cold standby component activated by another failure**
 - **A system configuration change due to a failure**
 - **A system configuration change responding to a signal**
 - **Failures that occur in a particular sequence**
 - **Failure criteria that change for a new mission phase**



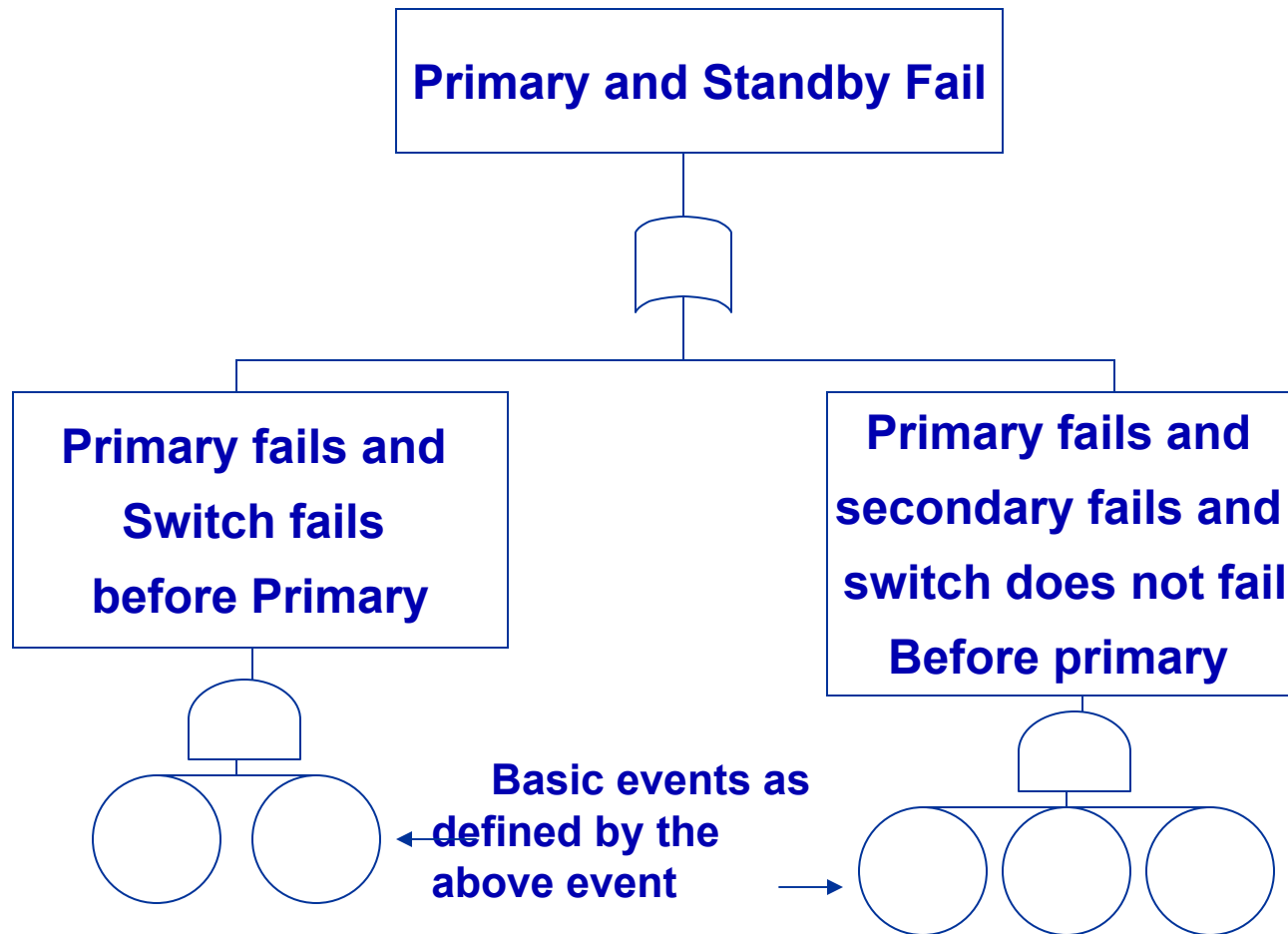
Example of a Dynamic System



After Primary failure switch to Standby



Outline of the FT for the Dynamic Example





Dynamic Events Can Be Handled by FTA

- Each event is clearly described to include the dynamic conditions
- The basic events are defined including the dynamic conditions
- Standard AND and OR gates are used to describe the general relational logic
- The difference is that more complex quantification formulas are used to incorporate the dynamic conditions



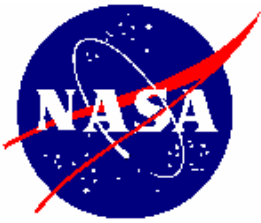
Special DFTA Software Can Be Used to Expedite the FTA

- **When there are numerous or complex dynamics, special DFTA software can be used**
- **The DFTA software incorporates special gates to show standby relations, a common supply, sequential relations, or re-configurations**
- **Markov analysis is used to quantify the dynamic events**



DFTA Exercise

- **Assume two processors share a common cold spare**
- **Develop the fault tree logic structure for the top event : No Processing Capability**
- **Determine the resulting minimal cutsets**
- **Discuss how the minimal cutsets would be quantified**



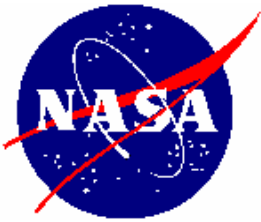
Applications of FTA Revisited

- **Understanding of System Failure and Contributors**
- **Identification of Design Features and Weaknesses**
- **Evaluation of Tradeoffs**
- **Prioritization of Contributors to Focus Actions**
- **Comparison with a Goal**
- **Minimization of Failure Probability**
- **Diagnosing Causes of a Failure or an Incident**



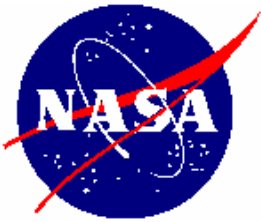
The Use of FTA to Understand System Failure and its Contributors

- **The FTA logically traces a system failure to its immediate causes**
- **These immediate causes are traced to their immediate causes, etc., until the basic component failure causes are identified**
- **This tracing of causes lays out the failure logic of the system in terms of causal failures**
- **A complete system failure mapping is thus obtained**



FTA: Understanding/Communicating

- Formal documentation of the system failure analysis
- A structured tool for what-if analysis
- A pictorial of failure progression paths to system failure
- A failure diagram of the system to be maintained with the system drawings
- A tool to extract information to communicate with engineers, managers, and safety assessors



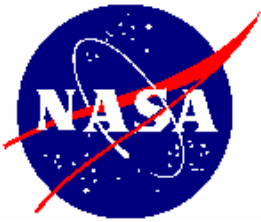
The Use of FTA to Identify Design Features and Weaknesses

- A single component minimal cutset identifies a single event or single failure that can cause the top event
- A minimal cutset containing events which are of all the same type has susceptibility to a single common cause triggering the events
- Minimal cutsets of significantly different size show potential system unbalances
- Minimal cutsets grouped according to given features show corresponding design features



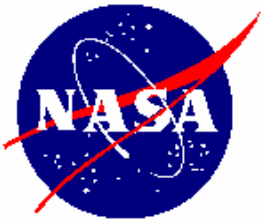
The Fault Tree as a Master Logic Diagram

- **The Master Logic Diagram (MLD) is a fault tree identifying all the hazards affecting a system or mission**
- **The Master Logic Diagram can also be called a Master Hazards Diagram (MHD)**
- **The MLD or MHD is developed using fault tree logic**
- **The basic events of a system MHD are the hazards that can initiate component failures or increase their likelihood**
- **The basic events of a mission MLD are the hazards that are the initiating events of potential accident scenarios**



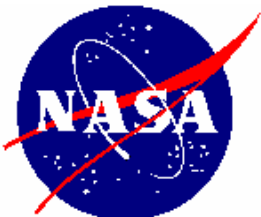
The MLD Identified the Initiating Events in the Space Shuttle PRA

- **The top event was Loss of Crew and Vehicle (LOCV)**
- **LOCV was resolved into mission phase contributions**
- **Each mission phase contribution was resolved into system contributors**
- **Each system contributor was resolved into initiating event contributors**

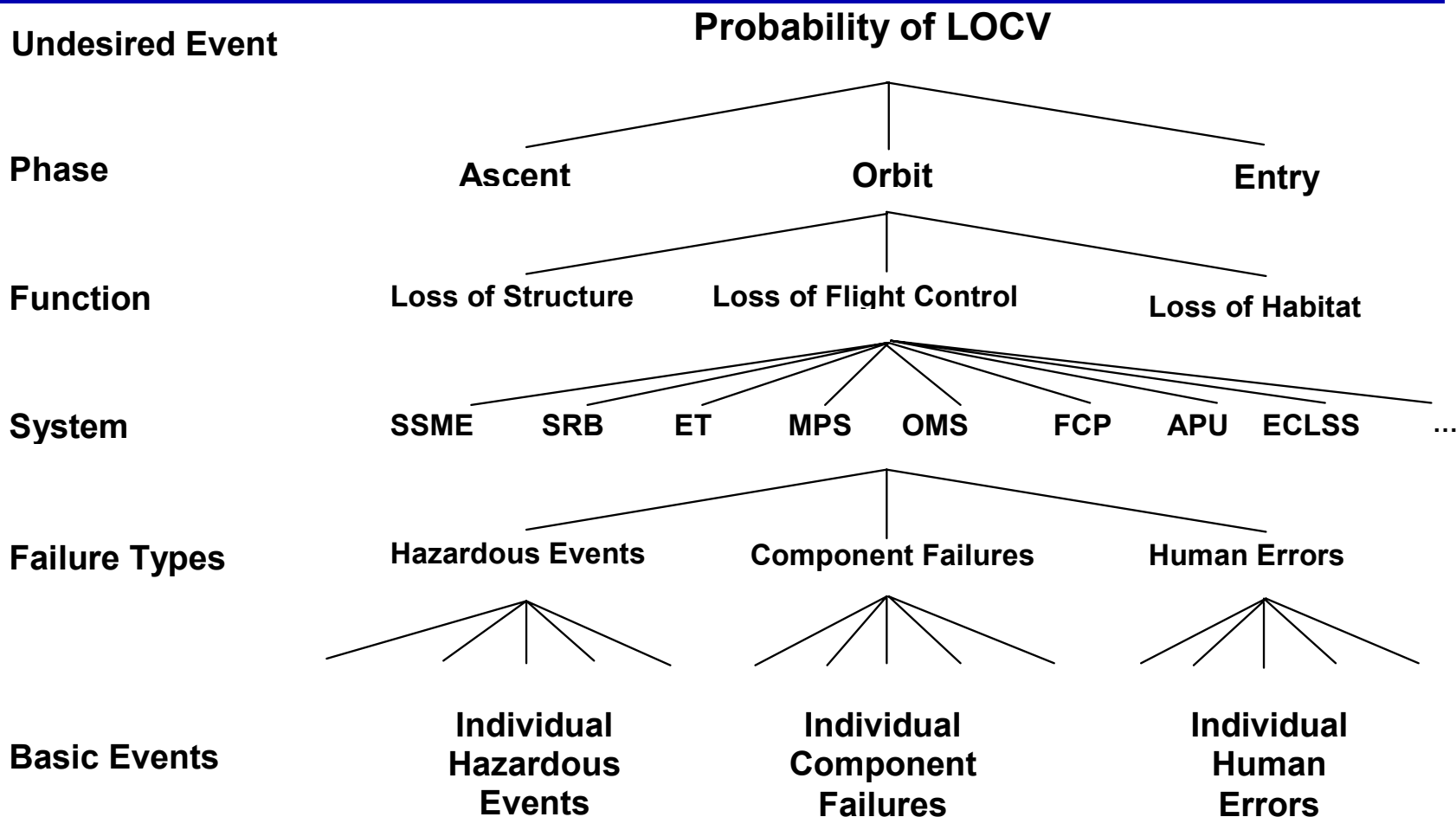


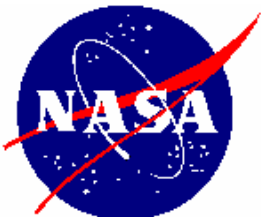
Dispositioning of the Initiating Events in the PRA

- **The initiating events were labeled**
- **Each initiating event was cross-referenced to hazards identified in Hazard Analyses**
- **Events were modified to be consistent with the Hazard Analyses**
- **Each event was dispositioned as to where it is modeled or if not modeled then why**

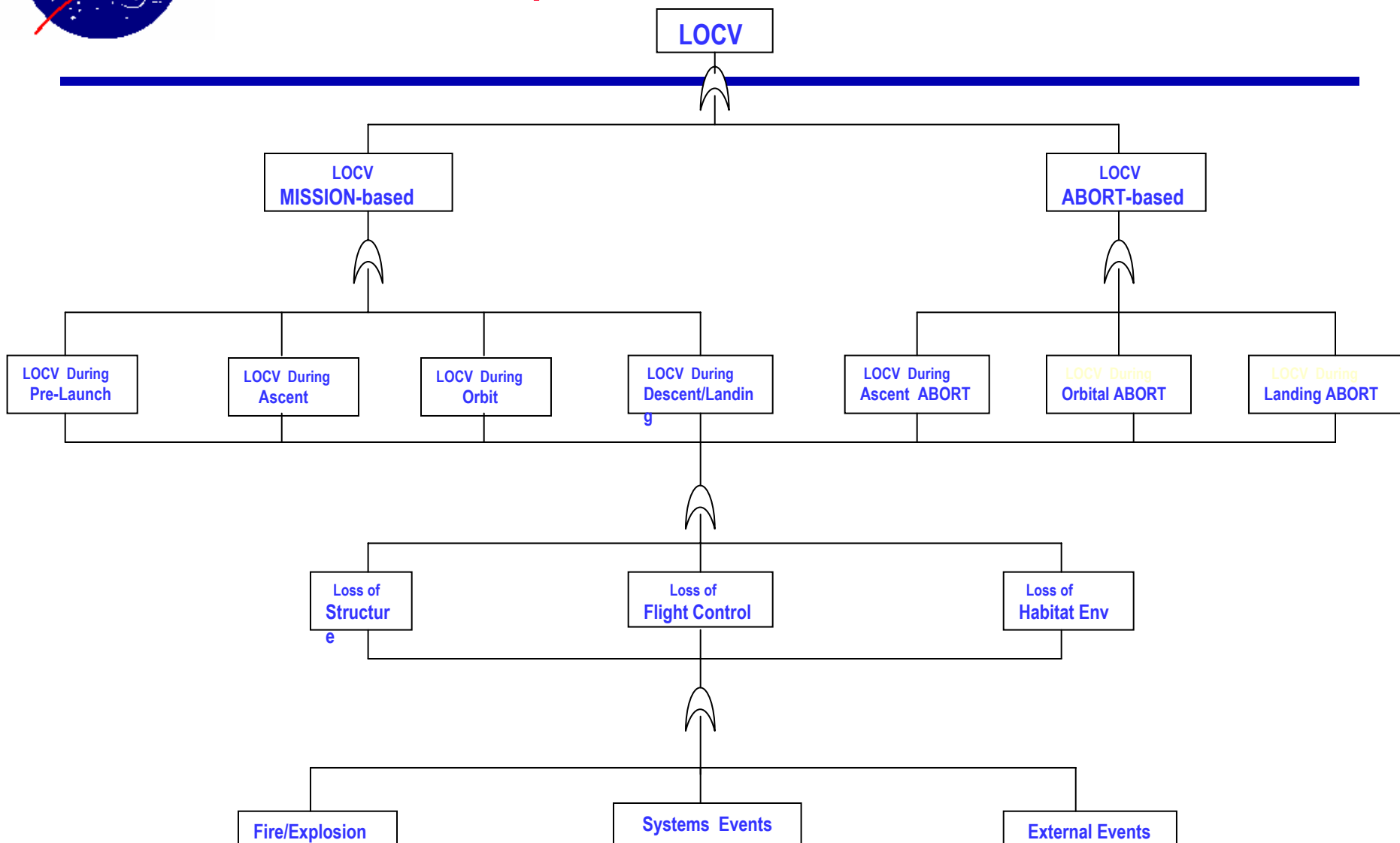


Structure of the MLD for the Space Shuttle PRA





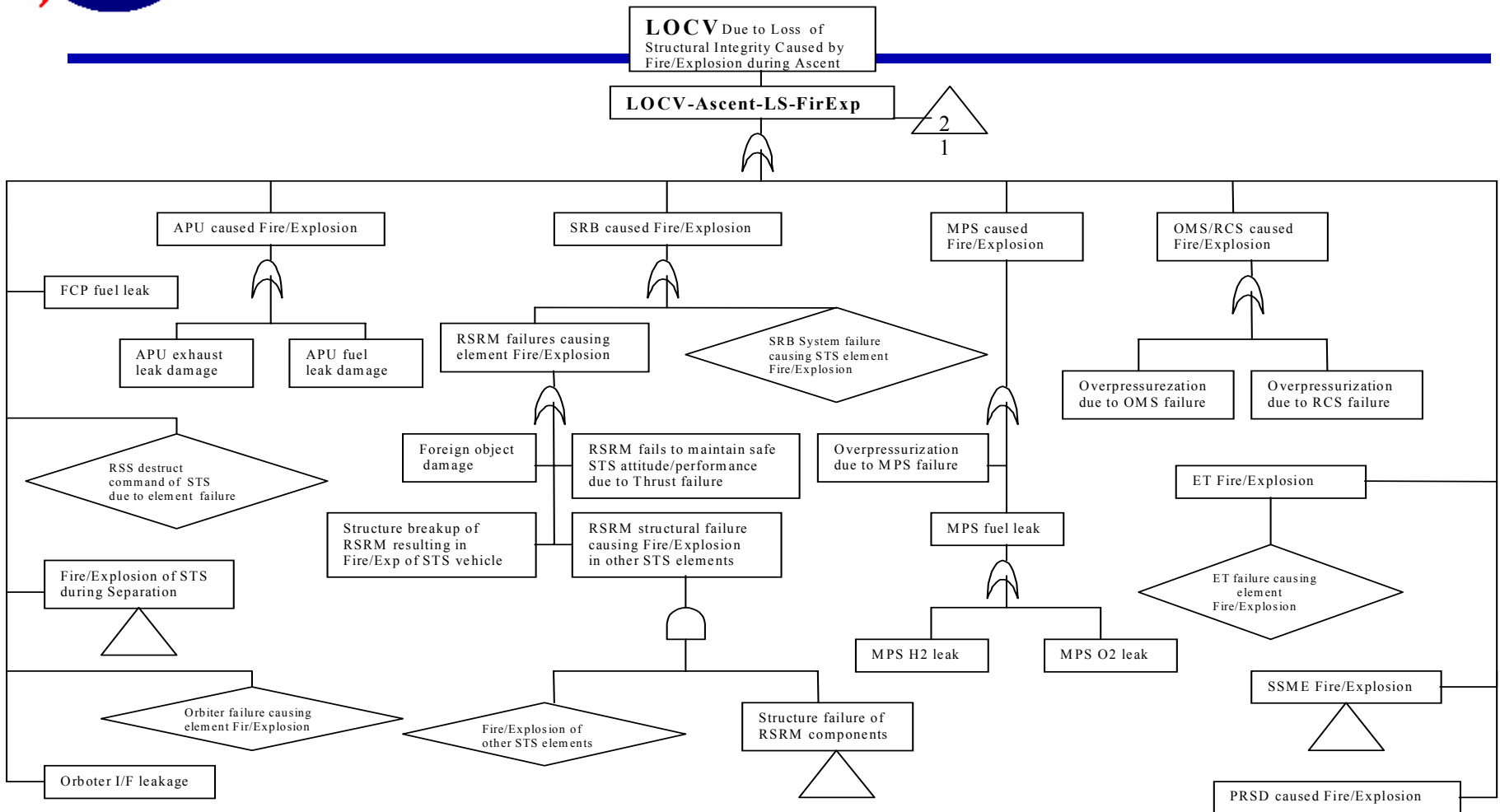
The Space Shuttle MLD Continued





Mission Success Starts With Safety

Further Development of the IE-MLD for Fires and Explosions on Ascent





Mission Success Starts With Safety

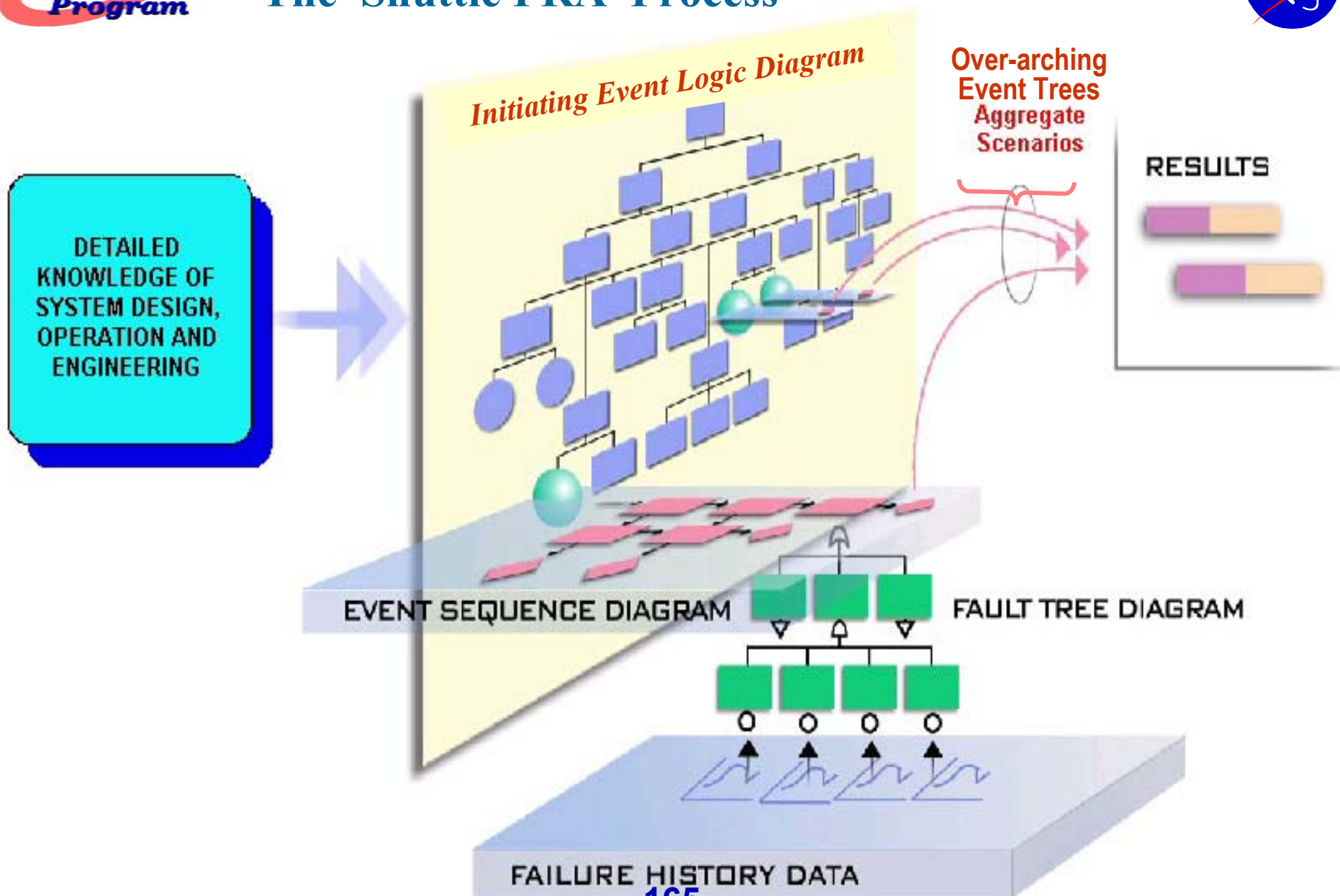
Cross-Reference of Hazard Reports with MLD Events

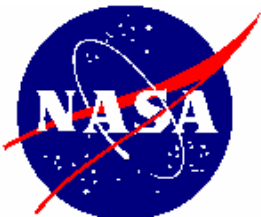
USA Hazard Number	MLD initial event	Mission Phase	System	PRA Consequence	Threatened Function			Hazard		Probability	
								F/P	Type	Sev	Like
ORBI 275	184	PAOD	ECLSS	LOCV		FC	HE	P	EE	A	b
ORBI 339	221	D	ECLSS	LOCV			HE	P	SE	A	c
ORBI 511	231	AOD	ECLSS	LOCV			HE		SE	A	c
ORBI 117	135	PAOD	ECLSS	LOCV		FC	HE	P	SE	A	d
ORBI 241	170	PAOD	ECLSS	LOCV			HE	P	SE	A	d
ORBI 321	208	D	ECLSS	LOCV			HE	P	SE	A	d
ORBI 254	176	O	ECLSS	Abort			HE	P	SE	B	d
ORBI 276	185	PAOD	ECLSS	Abort			HE	F	EE	B	d
ORBI 323	210	O	ECLSS	Abort			HE	P	SE	B	d

List of Accident Initiating Events Identified in the IMLD (*MPS Related Initiators*)

USA Hazard Num ber	MLD initia even t	Missi on Phas e	System	PRA Consequ e	Threatene d Function			Hazard Category		Prob Category		Referen ce ESD Names	Analyst Remarks		Individual Hazard Description
								F/P	Type	Sev	Like		FT/E T	Justi ficati on	
INTG 006	4	PA	MPS	LOCV	SI			P	FE	A	c				Ignition of Flammable Atmosphere at the ET / Orbiter LH2 Umbilical Disconnect Assembly
INTG 009	6	P	MPS	LOCV	SI	FC	HE	F	FE	A	c				Isolation of the ET from the Orbiter MPS or SSMEs (17 inch valve bursts open under pressure from ET)
INTG 016	12	PA	MPS	LOCV	SI	FC		P	FE	A	c				Ignition Sources Igniting Flammable Fluids in the Aft Compartment
INTG 019	390	A	MPS	LOCV		FC		F	SE	A	c				Premature shutdown of one or more SSME's
INTG 020	18	A	MPS	LOCV	SI	FC		P	FE	A	c				Hydrogen Accumulation in the Aft Compartment During Ascent
INTG 023	20	A	MPS	LOCV	SI	FC		P	FE	A	c				Contamination in the Integrated Main Propulsion System (which clogs the system)
INTG 034	24	PA	MPS	LOCV	SI	FC		P	FE	A	c				Autoignition in High Pressure Oxygen Environment (in MPS)
INTG 041	392	PA	MPS	LOCV		FC		F	FE	A	c				Loss of MPS/SSME He supply pressure
INTG 042	32	PA	MPS	LOCV	SI			P	SE	A	c				Turbopump Fragmentation During Engine Operation
INTG 112	48	AD	MPS	LOCV	SI	FC		P	FE	A	c				H2/O2 Component Leakage During Ascent/Entry
INTG 112	49	AD	MPS	LOCV	SI	FC		P	FE	A	c				H2/O2 Component Leakage During Ascent/Entry
INTG 168	81	PA	MPS	LOCV	SI	FC			EE	A	c				Flammable Atmosphere in the ET Intertank (see 238)
ORBI 035	102	AD	MPS	LOCV	SI	FC		P	FE	A	c				Hydrogen Accumulation in the Orbiter Compartments During RTLS/TAL Abort
ORBI 045	107	PAOD	MPS	LOCV	SI	FC	HE	P	FE	A	c				Ignition of Orbiter Fluids Entrapped in the TCS Materials (aft compartment)
ORBI 108	133	PAOD	MPS	LOCV	SI			P	SE	A	c				Overpressurization of the Orbiter Aft Fuselage Caused by the Failure of an MPS Helium Regulator or Relief Valve
ORBI 278	187	PAOD	MPS	LOCV	SI			P	SE	A	c				Loss of Structural Integrity Due to Overpressurization of the Mid and/or Aft Fuselage
ORBI 306	205	PA	MPS	LOCV	SI	FC		P	FE	A	c				Fire/Explosion in the Orbiter Aft Compartment Caused by MPS Propellant Leakage / Component Rupture
ORBI 338	219	PA	MPS	LOCV	SI	FC		P	FE	A	c				GO2 External Tank Pressurization Line as MPS/APU Ignition Source
ORBI 343	224	PA	MPS	LOCV	SI	FC		P	FE	A	c				Fire/Explosion in the Orbiter Aft Compartment Caused by Contamination in the Main Propulsion System Feed System
INTG 085	44	P	MPS	LOCV	SI			P	FE	A	d				Ignition of Flammable Atmosphere at T-0 Umbilicals
INTG 089	45	PA	MPS	LOCV	SI			F	SE	A	d				Malfunction of the LH2 and LO2 T-0 Umbilical Carrier Plate Resulting in Damage to Shuttle Vehicle
INTG 153	71	P	MPS	LOCV	SI			P	EE	A	d				Potential Geysering in the LO2 Feed Line (Tsat = boiling point)
INTG 166	79	P	MPS	LOCV	SI	FC		P	SE	A	d				Premature Separation of Orbiter T-0 Umbilical Carrier Plate
INTG 167	80	P	MPS	LOCV	SI	FC		P	SE	A	d				Overpressurization of LO2 Orbiter Bleed System or LH2 Recirculation System
ME-FG3P	346	PA	MPS	LOCV	SI			P	SE	A	d				geysering of LOX (MPS) (see 71)
ME-FG6S	354	P	MPS	LOCV	SI			P	SE	A	d				abnormal thrust loads
ME-FG8M	356	A	MPS	LOCV	SI			P	SE	A	d				thrust oscillations leading to pogo (see 3)
ORBI 248	172	PAOD	MPS	LOCV	SI	FC		P	FE	A	d				Fire/Explosion in GOX Pressurization System
ME-FA1S	310	P	MPS		SI	FC			FE	C	c				hydrogen fire/explosion external to aft compartment (see 21)

The Shuttle PRA Process

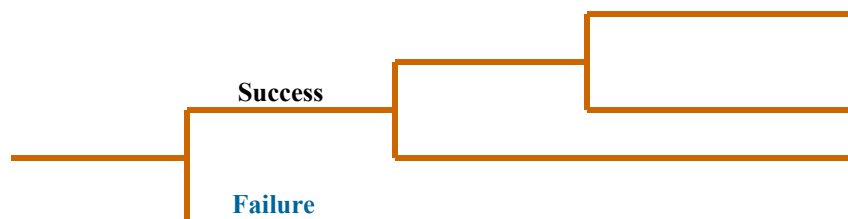




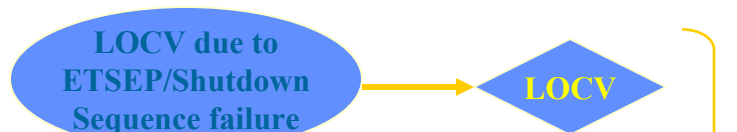
ET-SEP/MPS Shutdown Accident Sequences

System/Element Level Model Integration

PROPL-OK MECO ET-SEP MPS-DUMP END-STATES Freq.



OK		XXXXX
LOCV-DMP	1	XXXXX
LOCV-ETSEP	2	XXXXX
LOCV-MECO	3	XXXXX



FT for Top Event #5 Identified in Over-Arching Mission Model





Extending a System Fault Tree to a Master Hazard Diagram (MHD)

- The top event is defined as a system failure event
- The fault tree is developed to the basic component level
- Each component failure is further resolved into hazards and conditions that can cause failure or increase its likelihood
- The resulting system MHD identifies the hazards affecting the system and their consequences
- Of particular importance are single failures and hazards affecting multiple redundant components



Ranking the Criticality of Hazards Using FTA

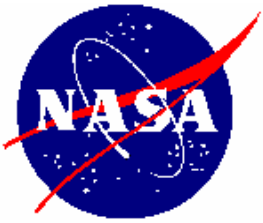
- Each hazard is linked to a basic event or events on the fault tree
- Equivalently each hazard is linked to the basic events in the minimal cutsets
- The criticality of the hazard is the likelihood of the hazard times the importance of the basic event
- The component importance is determined from the FTA
- The likelihood is determined from the hazard analysis

**Hazard Criticality=Likelihood x Importance of
Components Affected**



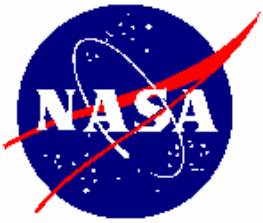
The Role of FTA in Mishap Analysis

- The accident scenario is constructed for the mishap
- System failures (pivotal events) are identified which resulted in the mishap
- A fault tree is constructed for each system failure to resolve the basic events involved
- For further root cause analysis a basic event is resolved into the possible causes
- The basic events (or root causes) are dispositioned according to their plausibility or likelihood

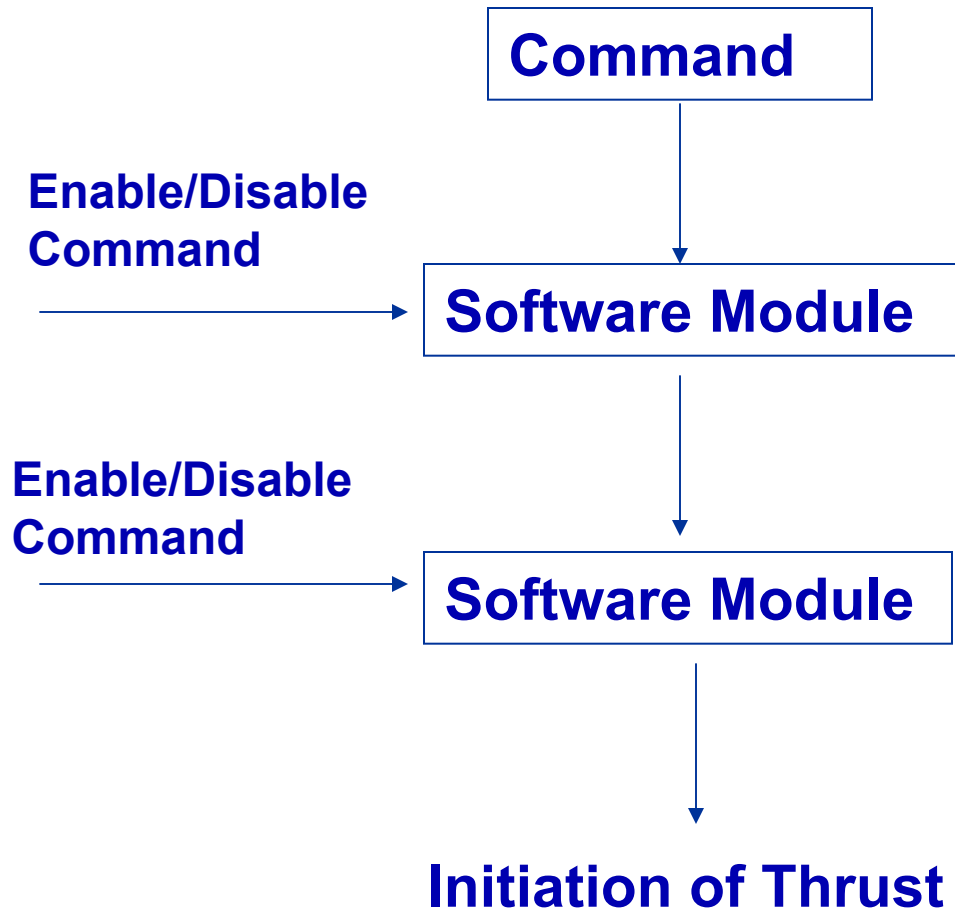


FTA Applied for Software Assurance

- FTA can be applied to a software program to analyze the logic flow
- FTA can be applied to software coding to analyze detailed command and data transmittal
- The same FT process as applied to hardware is applied to software
- A top event defines a particular software undesired output or lack of output
- The top event is resolved into immediate, necessary and sufficient events for the top event
- The resolution is traced back to software failures or input failures



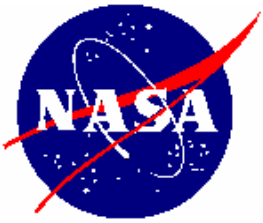
The Equivalent Monopropellant Software Diagram





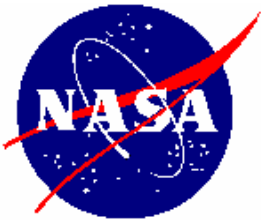
FTA in Design

- **Top level fault trees are developed**
 - **Functional level**
 - **System level**
 - **Subsystem level**
- **Tradeoffs are carried out**
 - **Alternative functional capabilities**
 - **Alternative redundancies**
- **Allocations are performed**
 - **System requirement into subsystem requirements**
 - **Subsystem requirements into component requirements**



The Use of FTA to Evaluate Tradeoffs

- **Tradeoffs involve alternatives to design or operation**
- **FTA evaluates alternatives by appropriately modifying the FT**
- **Changes in the top event results show the impact of the alternatives**
- **The changes can be qualitatively or quantitatively evaluated**



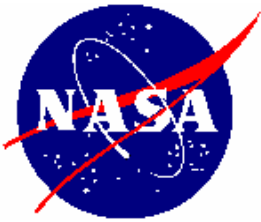
Monopropellant Design Tradeoff FTA

- What would the benefit be of adding an additional, redundant isolation valve in the fuel supply line?
- What is the effect of replacing the manual emergency switch S3 with an automatic timer relay?
- What is the effect of removing the automatic timer relay K6 circuit and having the relay K5 connect to S3 which now becomes an automatic timer?
- What is the effect of adding an additional timer relay as a redundancy to K6?



The Use of FTA to Prioritize Contributors

- **Each basic event in the fault tree can be prioritized for its importance to the top event**
- **Different importance measures can be obtained for different applications**
- **Basic events are generally significantly different in their importance providing effective prioritization**
- **In addition to the basic events, every intermediate event in the FT can be evaluated for its importance**



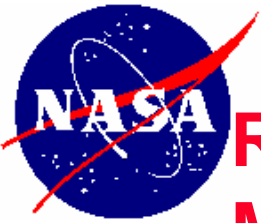
Use of FTA to Compare with a Goal

- **FTA can be used to calculate a top event probability that can be compared with a goal**
- **Uncertainty analysis can be incorporated by assigning each basic event an uncertainty distribution**
- **If the FTA is carried out according to defined ground rules and meaningful data are available then this can be meaningful**



Use of FTA in Minimizing Failure Probability

- **The fault tree equations can be programmed to handle different values for the failure probabilities, failure rates, and repair times**
- **Cost equations or resource equations can be included to handle these constraints**
- **The probability of system failure (represented as the top event) can be optimized using available software packages**



Reducing the Probability of the Monopropellant Failure to Terminate Thrust

- **What are the options for reducing the probability of failure to terminate thrust in the monopropellant example?**
- **How do these options effect the probability of no thrust for the other monopropellant example?**
- **Are there options which reduce both probabilities?**
- **What criteria can be used to determine whether such reductions are needed or are effective?**



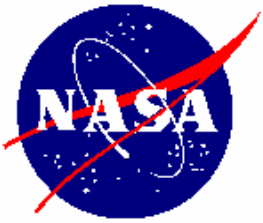
Use of FTA to Diagnose Causes of a Failure

- **FTA can also be used as a reactive tool to assess the causes of a failure**
- **The observed failure is the top event**
- **The FT is developed to identify the possible basic causes**
- **The basic causes can be prioritized for their likelihood using FT importance measures**



Diagnostic FTA

- The observed failure (end state) is the top event
- Observed successes and failures of subsystems and components are documented
- The top event is developed to the immediate possible causes
- Failures which cannot occur because of the observations are truncated and not further developed
- Tests are identified to resolve whether additional failures have occurred or have not occurred
- The FT is developed in this manner to resolve the plausible causes of the top event



Monopropellant Diagnostic FTA

- **Observed System Failure: Thruster Supplied with Propellant after Thrust Cutoff**
- **Additional Observed Events: No continued EMF measured in any of the circuits**
- **Diagnostic FT: All continued EMF events deleted from the original FT**
- **The basic causes identified are Isolation Valve IV3 and Isolation Valve IV2 failures**
- **If the diagnostic FT was developed after the observed event then no EMF events would be further developed and would be nullified**



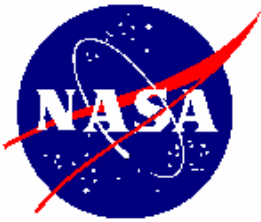
The Mirror Success Tree (ST)

- A Success Tree (ST) identifies all the ways in which the top event *cannot* occur
- The ST is the *complement* of the FT
- The ST is the *mirror* of the FT
- The ST is useful in showing the explicit ways to *prevent* the occurrence of the FT
- The ST is the *success space* twin of the FT
- The ST does not as clearly differentiate importances and priorities for preventing the top event
-



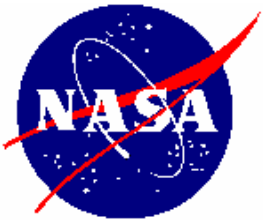
Determining the ST from the FT

- Complement the top event to a NOT event
- Complement all intermediate events to NOT events
- Complement all basic events to NOT events
- Change all AND gates to OR gates
- Change all OR gates to AND gates
- The tree is now the ST
- The minimal cut sets of the ST are now called the minimal path sets

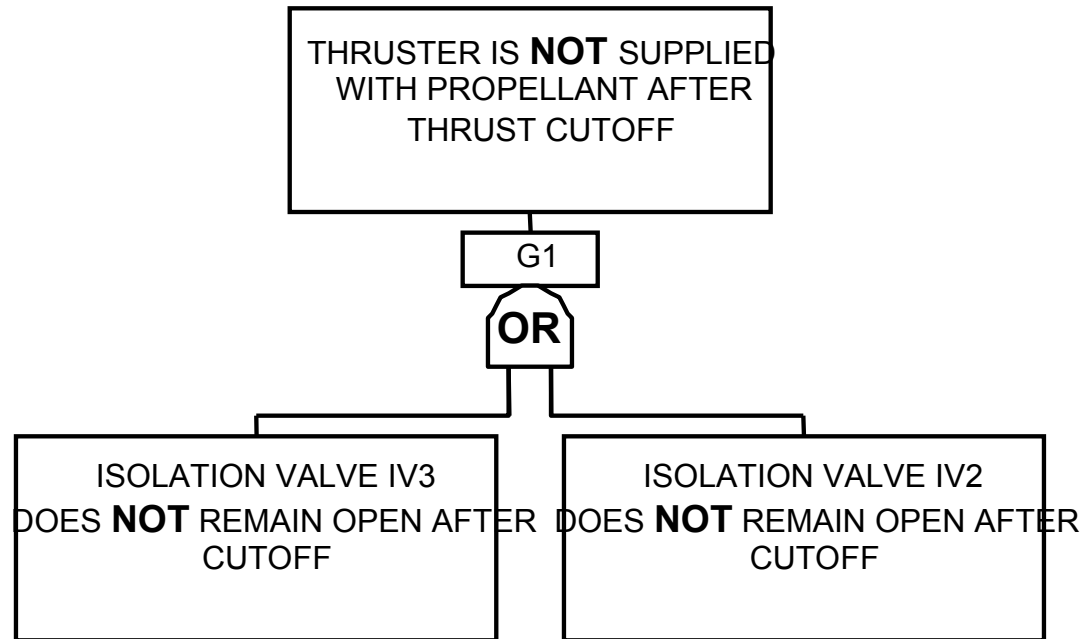


Minimal Path Sets

- A minimal path set is the smallest number of events which if they all do not occur then the top event will not occur
- If the events in one path set are prevented to occur then the top event will be guaranteed to not occur
- The minimal path sets are the totality of ways to prevent the top event based on the fault tree



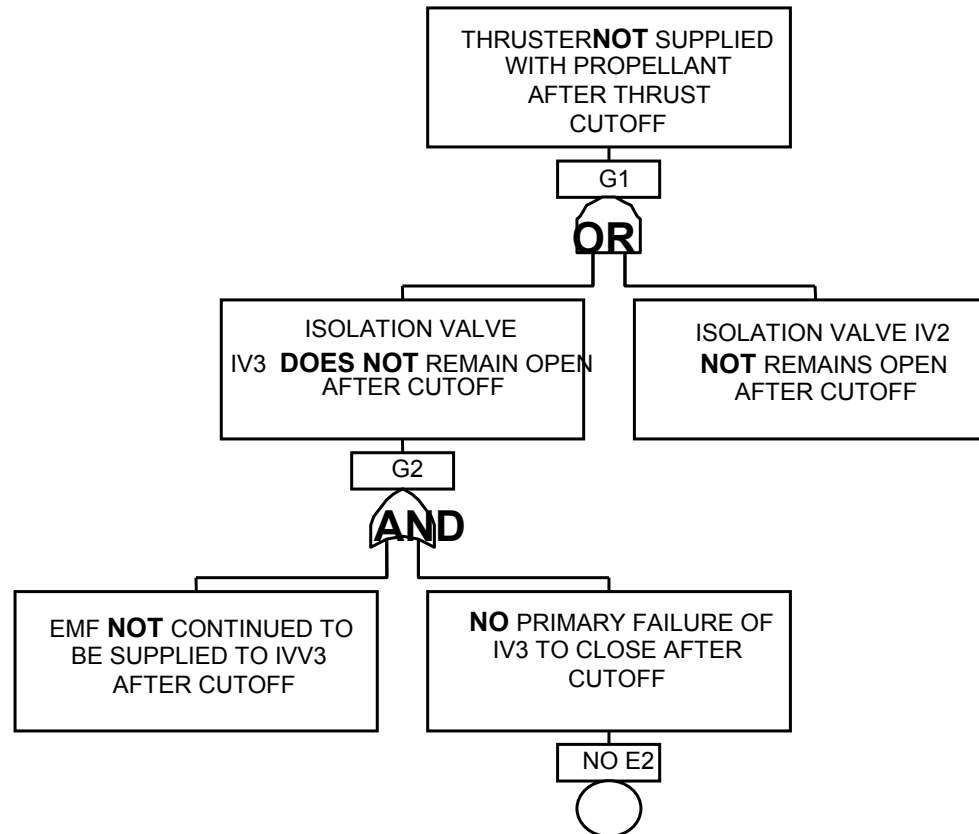
Top Part of Monopropellant Success Tree



Success Tree Construction – Step 1



Success Tree Construction – Step 2





Minimal Path Sets from the Minimal Cut Sets

- Take the complement of the union of the minimal cut sets (mcs)
- Carry out Boolean manipulation to obtain a union of intersections
- The intersections, or combinations of events, are the minimal path sets (mps)
- The set of minimal path sets is the totality of combinations of preventions stopping the top event from occurring



Monopropellant FT: MPS from MCS

$$T = E6E7 + E6E8 + E5E7 + E5E8 + E1E3 + E1E4 + E1E2$$

Take the complement (denoted by a superscript):

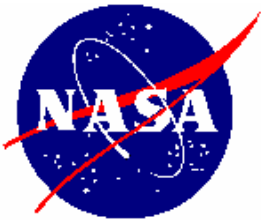
$$T' = (E6E7 + E6E8 + E5E7 + E5E8 + E1E3 + E1E4 + E1E2)'$$

Apply the Union Complement Law

$$T' = (E6E7)'(E6E8)'(E5E7)'(E5E8)'(E1E3)'(E1E4)'(E1E2)'$$

$$T' = (E6' + E7')(E6' + E8')(E5' + E7')(E5' + E8')(E1' + E3')(E1' + E4')(E1' + E2')$$

$$T' = E6'E5'E1' + E7'E8'E1' + E6'E5'E3'E4'E2' + E7'E8'E3'E4'E2'$$



FTA Interface with Reliability Analysis

- For quantification, the basic component inputs to FTA are component failure rates and repair rates
- For a first order calculation, the failure rates and repair rates are treated as being constant
- For more detailed quantifications, the failure rates and repair rates can be modeled as being age or time dependent
- Weibull distributions are often used for the failure times
- Lognormals or threshold exponential can be used for the repair times
- FTA can be linked to failure and repair data records



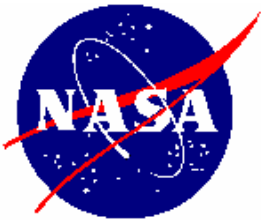
FTA Project Management Tasks (1)

- **Define the FTA**
 - Top Event
 - Scope
 - Resolution
- **Assemble the project Team**
 - FT analyst
 - System engineering support
 - Data support
 - Software support
- **Define the FTA Operational Framework**
 - Assemble the as built drawings
 - FT naming scheme
 - Interfaces/Support to be modeled
 - Software to be used



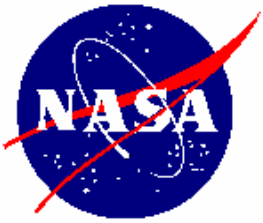
FTA Project Management Tasks (2)

- **Assemble the data**
 - Generically applicable data
 - Specifically applicable data
- **Prepare the software package**
 - Familiarization
 - Test problems
- **Keep a log on the FTA work**
 - Operational and design assumptions
 - Events not modeled and why
 - Success and failure definitions
 - Special models and quantifications used



FTA Project Management Tasks (3)

- **Review the work at stages**
 - FT construction
 - Qualitative evaluations
 - Quantitative evaluations
- **Check and validate the results**
 - Engineering logic checks
 - Consistency checks with experience
- **Prepare and disseminate the draft report**
 - Conclusions/findings
 - FTA results
 - FTs
 - Software inputs/outputs
- **Obtain feedback and modify and final report**
 - Disseminate the report
 - Present findings



Reference

- **“Fault Tree Handbook with Aerospace Applications’,
Version 1.1, NASA Publication, August 2002.**